3V700D

400MHz FSB / VGA / SATA / LAN /

3.5" board size

400MHz FSB . All-in-one .
Sound . LAN . PCIE Mini Card.
SATA . USB . CF . 12V DC-IN

NO. 3V700D
Release date: OCT . 27. 2008

Contents

3V700D
Warning!..... 1
Hardware Notice Guide 2
CHAPTER 1 GENERAL INFORMATION 4
1-1 MAJOR FEATURE..... 5
1-2 SPECIFICATION 6
1-3 INSTALLING THE SO-DIMM 7
1-3-1.1 REMOVING THE SO-DIMM 9
1-4 PACKING LIST 10
CHAPTER 2 HARDWARE INSTALLATION 12
2-1 UNPACKING PRECAUTION 12
2-2 UNPACKING CHECKUP 13
2-3 3V700D-CRXXX DIMENSION 14
2-3-1 3V700D-VRXXX DIMENSION 15
2-4 LAYOUT 16
2-5 3V700D-CRXXX DIAGRAM 17
2-5-1 3V700D-VRXXX DIAGRAM 18
2-6 INSTALL MEMORY 19
2-7 LIST OF JUMPERS 20
2-8 JUMPER SETTING DESCRIPTION 21
2-9 CMOS DATA SET 22
2-10 JUV1: USB VOLTAGE SELECT 23
CHAPTER 3 CONNECTION 24
3-1 LIST OF CONNECTORS..... 24
3-2 FAN CONNECTOR 25
3-3 SERIAL ATA..... 26
3-4 I^2C BUS CONNECTOR (OPTIOMAL) 26
3-5 COMPACT - FLASH MEMORY SOCKET..... 27
3-6 COM PORT CONNECTOR 28
3-7 LAN PORT 29
3-8 USB HEADER 30
3-9 VGA PORT HEADER 32
3-10 PS2 KEYBOARD /MOUSE WAFER CONNECTOR 33
3-11 AUDIO WAFER CONNECTOR 33
3-12 DC 12V-IN POWER JACK 34
3-13 DC +5/+12V OUTPUT WAFER CONNECTOR 34
3-14 FORNT PANEL WAFER CONNECTOR 35
3-15 POWER BUTTON WAFER CONNECTOR 35
3-16 LAN LED WAFER CONNECTOR 35

3-17 FOR F75111N I2C WATCH DOG TIMER DEVICE 36
3-18 I2C WDT/DIO W75IO.H REFERENCE SAMPLE CODE 37
3-18-1 I2C WDT/DIO F75111N REFERENCE SAMPLE CODE 41
CHAPTER 4 INTRODUCTION OF BIOS..... 51
4-1 ENTER SETUP 51
4-2 GETTING HELP 52
4-3 THE MAIN MENU 52
4-4 STANDARD CMOS FEATURES 54
4-5 ADVANCED BIOS FEATURES 55
4-6 ADVANCED CHIPSET FEATURES 56
4-6-1 DRAM CLOCK / DRIVE CONTROL 56
4-6-2 AGP & P2P BRIDGE CONTROL 57
4-7 INTEGRATED PERIPHERALS 57
4-7-1 VIA OnCHIP IDE DEVICE 58
4-7-2 VIA OnCHIP PCI DEVICE 58
4-7-3 SUPER IO DEVICE 59
4-8 POWER MANAGEMENT SETUP 59
4-8-1 IRQ / EVENT ACTIVITY DETECT 60
4-8-1.1 IRQS ACTIVITY MONITORING 60
4-9 PnP / PCI CONFIGURATION SETUP 61
4-10 PC HEALTH STATUS 62
4-11 FREQUENCY / VOLTAGE CONTROL 62
4-12 LOAD STANDARD/OPTIMIZED DEFAULTS 63
4-13 SET SUPERVISOR / USER PASSWORD 63
CHAPTER 5 DRIVER INSTALLATION..... 65
5-1 VIA 4 IN 1 INSTALL VIA IDE/AGP/ INF DRIVER 67
5-2 VGA INSTALL CN700 VGA DRIVER 69
5-3 SOUND INSTALL VIA AUDIO CODEC DRIVER FOR VIA 70
5-4 USB2.0 INSTALL VIA USB2.0 INSTALLING DRIVER..... 71
5-5 HOW TO DISABLE ON-BOARD SOUND 72
5-6 HOW TO UPDATE BIOS 72
APPENDIX A: POWER CONSUMPTION TEST..... 73
APPENDIX B: RESOLUTION LIST 74

Copyright

This manual is copyrighted and all rights are reserved. It does not allow any non authorization in copied, photocopied, translated or reproduced to any electronic or machine readable form in whole or in part without prior written consent from the manufacturer.

In general, the manufacturer will not be liable for any direct, indirect, special, incidental or consequential damages arising from the use of inability to use the product or documentation, even if advised of the possibility of such damages.

The manufacturer keeps the rights in the subject to change the contents of this manual without prior notices in order to improve the function design, performance, quality, and reliability. The author assumes no responsibility for any errors or omissions, which may appear in this manual, nor does it make a commitment to update the information contained herein.

Trademarks

Intel is a registered trademark of Intel Corporation.
Award is a registered trademark of Award Software, Inc.
All other trademarks, products and or product's name mentioned here are for identification purposes only, and may be trademarks and/or registered trademarks of their respective companies or owners.

ⓒCopyright 2008
All Rights Reserved.
User Manual edition 0.1, OCT . 27. 2008

Warning !

1. CF card & 2.5" HDD first boot issue
Due to the frequent update and change of Compact Flash card, it can't always work with 2.5" HDD at the same time. Users may try and get the test result.

2. Battery
Battery on board is consumables. Lex doesn't guarantee the life time of it.

3. Fanless solution with HDD
Please be aware of specification & limitation for HDD when fanless solution is implemented.

4. Lex will not give further notification if there is any change about the product information and the manual.

5. SATA does not support Hot SWAP

6. There would be ± 20% difference of WDT at room temperature.

7. Please make sure the voltage specification meet the requirement of the equipment before plugging into the power.

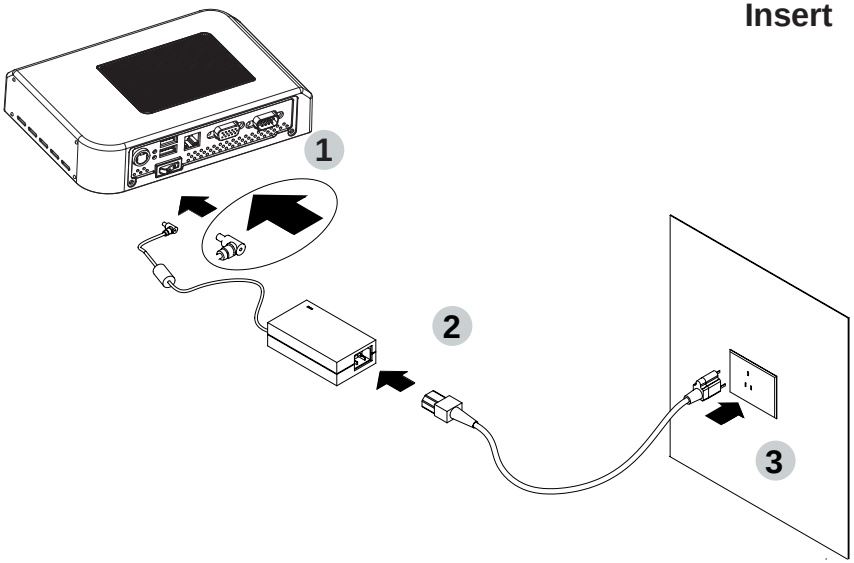
Hardware Notice Guide

1. Before installing the power supply with this motherboard, please attach the 12V/DC (2 pin connector)of the adapter to motherboard first.
   After that, plug the adapter power to AC outlet.
   Always normally shut down the computer before you move the system unit or remove the power supply from the motherboard.
   Please unplug the 12V/DC (2 pin connector) of the adapter from motherboard first.
   Then unplug the adapter from the AC outlet.
   Please refer to procedure from the photo 1
2. There will be high possibility to burn out the CPU if you change/ modify any parts of the CPU cooler.
3. Please wear wrist strap and attach it to a metal part of the system unit before handling a component.
   You can also touch an object that is of ground connection or with metal surface if you don't have wrist strap.
4. Please be careful when you handle this product. Pay attention to & don't touch the sharp-pointed components at the bottom PCB .
5. Please pay attention to this: Remove or change any components form the motherboard will VOID the warranty of the motherboard you purchased .
6. Before you install/remove any components or make any jumper setting on the motherboard, please make sure to disconnect the power first.
   (Please follow the instructions as of this guide)
7. Please only use single sided Mini PCI card, do not use the double sided Mini PCI card which is not suitable.
8. This does not support 16 bit mini PCI card
9. Please follow this instruction carefully when using the "POWERON after PWR-Fair" function.
   When the DC power adaptor runs out of power, unplug it from the DC current;
   when power returns plug it back in only after 5 seconds. If there is a power outage,
   unplug it from the AC current, when power returns plug it back in only after 30 seconds.
   Otherwise it will cause system locking or serious damage.

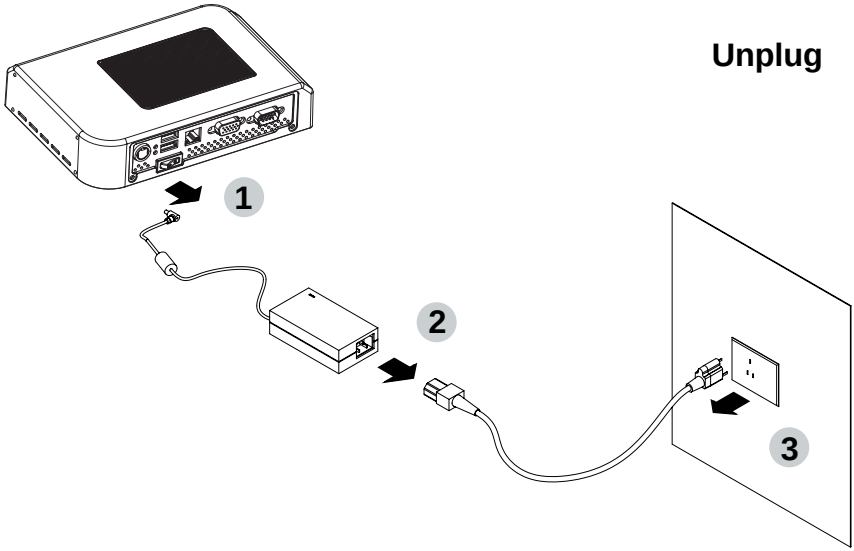
Remark 1:
Always insert/unplug the 12V/DC (2 pin connector) horizontally & directly from the motherboard.
DO NOT twist the 12V/DC (2 pin connector) gently, it is designed to fit snugly . Moreover, erratic pull / push testing with the DC Jack might cause the unpredictable damage to the component & system unit.

Photo 1

Insert



Unplug



Chapter-1

General Information

The 3V700D is a 3.5 inches form factor All-In-One control Board. The board design combines all necessary input and output effects interfaces, which makes itself an ideal all-in-one control board for Thin Client Application. The board is designed with 400 MHz internal bus clock rate architecture.

The 3V700D All-In-One motherboard uses VIA CN700 & VT8237R plus chipset, built-in VIA Eden(V4) /C7(V4) EBGA CPU, VGA and Audio feature onboard. This board offers the superb performance and PC specification in the industry.

The motherboard is fully compatible with industry standards, adding many technical enhancements, and is fully compatible with thousands of software applications developed for IBM PC/AT compatible computers. The control logic provides high-speed performance for the most advanced multi-user, multitasking applications available today.

A single Flash chip holds the system BIOS, and you can change the Flash BIOS by the Utility Update. Advanced IR port also provides a faster data transmission.

3V700D is designed with CN700 integrated graphic VGA controller which provides connection to VGA Monitor. The 3V700D Board supports DDR2 SDRAM memory with one unbuffered double-sided SODIMMs up to 1GB. One PCI Express Mini Card socket for USB device only is available for expansion purpose.

3V700D can be applied for Thin Client, Kiosk and Networking application.
The small size form factor allows itself to integrate as mobile PC or small panel PC.

1-1 Major Feature

1. VIA Eden (V4) nano BGA2 500MHz/1GHz CPU/ VIA C7 nano BGA2 1GHz CPU
2. VIA CN700 & VT8237R plus chipset
3. One DDR2 400/533 SDRAM unbuffered double-sided SODIMMs up to 1GB
4. 400 MHz system clock support
5. Versatile storage device:
   * One 50-pin Compact Flash socket
   * One SATA HDD (2nd SATA as option)
6. On board CN700 graphic controller integrated graphics
7. On board LAN 3 x Realtek 8100C 10/100 Mbps
8. Compact Flash Reader Type I/II for ATA interface
9. On board Sound, 2 Channels (Option)
10. On board mini PS/2 Keyboard/Mouse connector (Option)
11. 12V DC-IN Power support
12. Hardware watch dog timer (Option)
13. PCIe Mini Card socket for USB device only (Option)
14. 1 serial RS232 Ports (for 3V700D-CRXXX)

1-2 Specification

1. CPU: VIA Eden (V4) nano BGA2 500MHZ/1GHz CPU /
   VIA C7 (V4) nano BGA2 1GHz CPU
2. Cache Memory: Integrated full-speed 128KB L1/L2 cache
3. Chipset: VIA CN700 & VT8237R plus
4. Memory: One SODIMM socket up to 1GB DDR2 400/533 SDRAM
5. VGA: CN700 integrated graphic controller; memory sharable up to 64MB
6. I/O Chipset: VIA VT1211 IO
7. LAN: 3 x Realtek 8100C 10/100 Mbps (optional Realtek Gigabit)
8. Storage devices: * One 50-pin Compact Flash reader
   * One SATA HDD (2nd SATA as option)
9. USB: Support 8 USB internal headers
10. Keyboard/Mouse: PS/2 din connector (Option)
11. Sound: AC-LINK audio codec, 2 Channel Sound (Option)
12. BIOS: Award BIOS version 6.1
13. Dimension: 145x102mm
14. Power: 12V DC-IN power input
15. Power Voltage: +12V (11.4V to 12.6V)
16. Power Consumption: Please refer to page 73
17. Expansion: 1 x PCIE Mini Card socket for USB device only (Option)
18. Serial port: 1 x RS232 (for 3V700D-CRXXX)
19. WDT: Hardware watch dog timer, 0-255 second programmable (Option)

1-3 Installing the SO-DIMM

1. Align the SO-DIMM with the connector at a 45 degree angle.

2. Press the SO-DIMM into the connector until you hear a click.

Notices:

1.The connectors are designed to ensure the correct insertion. If you feel resistance, check the connectors & golden finger direction, and realign the card.

2. Make sure the retaining clips (on two sides of the slot) lock onto the notches of the card firmly.

1-3-1.1 Removing the SO-DIMM

1. Release the SO-DIMM by pulling outward the two retaining clips and the SO-DIMM pops up slightly.

2. Lift the SO-DIMM out of its connector carefully.

1-4 Packing List

3V700D-VRXXX

① User's Manual
② SATA DC Cable
③ SATA Data Cable
④ Adapter 30W
⑤ 3V700D-VRXXX Board
⑥ Utility CD Disk

3V700D-CRXXX

① 3V700D-CRXXX Board
② Adapter 30W
③ SATA Data Cable
④ VGA Cable
⑤ SATA DC Cable
⑥ User's Manual
⑦ Utility CD Disk

Optional

USB Cable

*The packing list above is for the users who purchase single motherboard. The users who purchase the board with chassis may refer to the packing list in the Assembly Guide.

Please contact with your dealer if any of these items is missing or damaged on delivery. And please keep all parts of the delivery package with packing materials in case if you need to deliver or store the product in the future.

Chapter-2

Hardware Installation

This chapter provides the information how to install the hardware of 3V700D. Please follow section 1-5, 2-1 and 2-2 to check the delivery package and unpack carefully. Please follow the jumper setting procedure.

2-1 Unpacking Precaution

The 3V700D board has been well packed with an anti-static bag to protect its sensitive components and circuitry from damage due to static electric discharge.

NOTE!

1. Do not touch the board or any other sensitive components without all necessary anti-static protection.
2. Please pay attention to the voltage limitation of DC-IN12 V ± 5 %.
   Overuse of DC-IN voltage limitation or change to another power adapter (not provided with this system) will VOID warranty.

You should follow these steps to protect the board from the static electric discharge whenever you handle the board:

1. Ground yourself by a grounded wrist strap at all times when you handle the 3V700D. Well secure the ALLIGATOR clip of the strap to the end of the shielded wire lead from a grounded object. Please put on and connect the strap before handling the 3V700D for harmlessly discharge any static electricity through the strap.

2. Please use anti-static pad to put any components, parts, or tools on the pad whenever you work on them outside the computer. You may also use the anti-static bag instead of the pad. Please ask your local supplier for necessary parts on anti-static requirement.

3. Do not plug any connector or set any jumper when the power is on.

2-2 Unpacking checkup

First of all, please follow all necessary steps of section 2-1 to protect 3V700D from electricity discharge. With reference to section 1-5, please check the delivery package again with following steps:

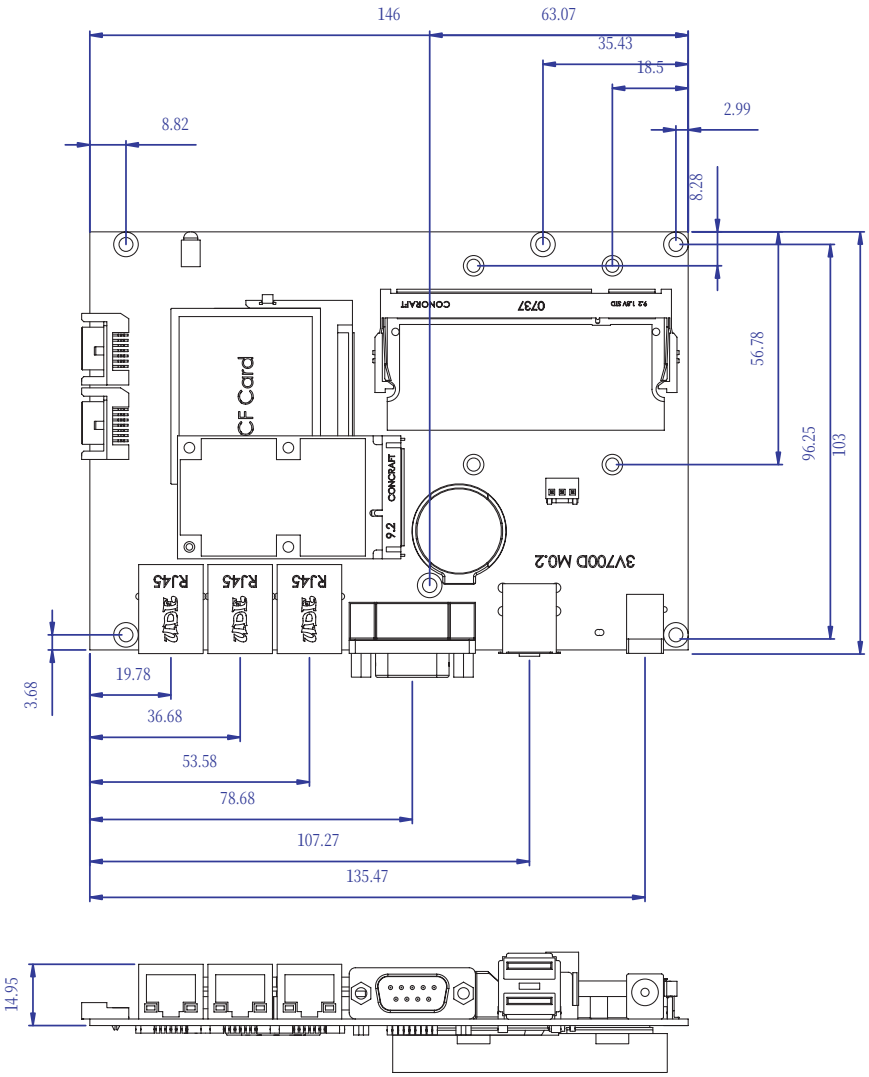
1. Unpack the 3V700D board and keep all packing material, manual and driver disc etc, do not dispose !

2. Is there any components lose or drops from the board? DO NOT CONTINUE TO INSTALL THIS BOARD! CONTACT THE DEALER YOU PURCHASED THIS BOARD FROM, IMMEDIATELY.
3. Is there any visible damage on the board? DO NOT CONTINUE TO INSTALL THIS BOARD!CONTACT THE DEALER YOU PURCHASED THIS BOARD FROM, IMMEDIATELY.
4. Check your optional parts (i.e. DDR2, CF etc.), all necessary jumpers setting to jumper pin-set, and CMOS setup correctly.
   Please also refer to all information of jumper settings in this manual.

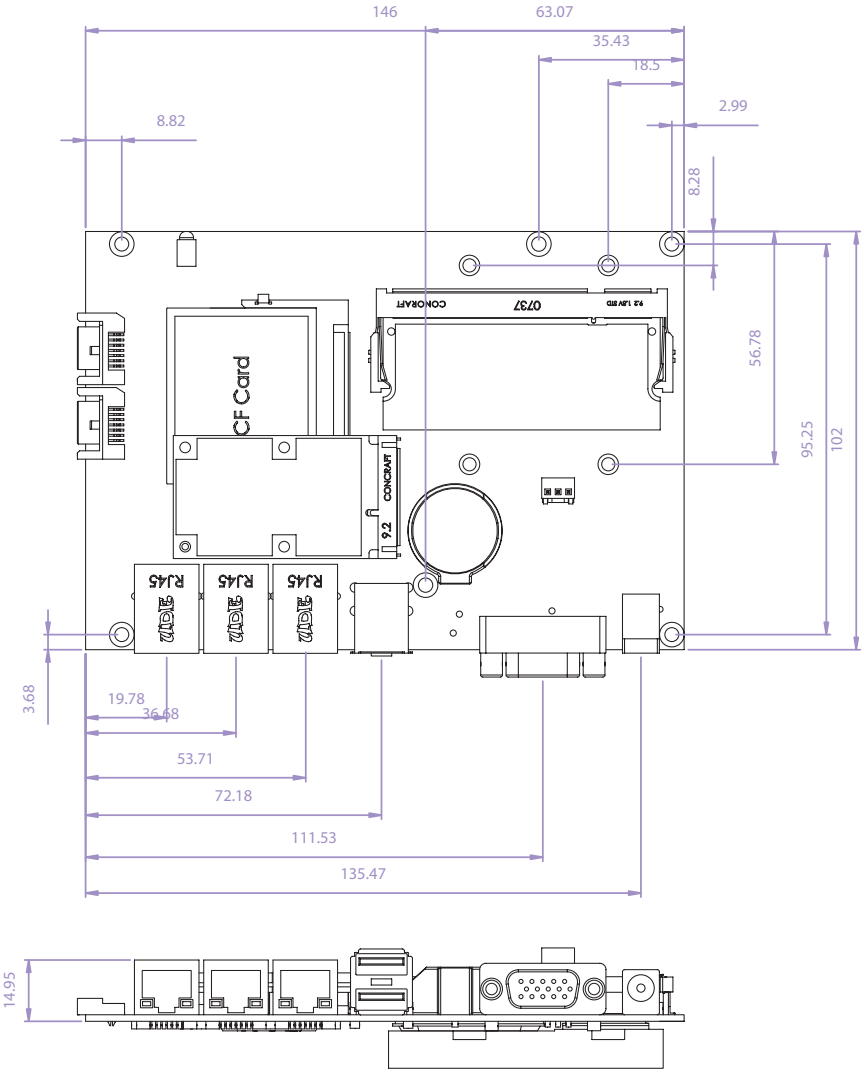
5. Check your external devices (i.e. Add-On-Card, Driver Type etc.) for complete add-in or connection and CMOS setup correctly.
   Please also refer to all information of connector connection in this manual.

6. Please keep all necessary manual and driver disc in a good condition for future re-installation if you change your Operating System.

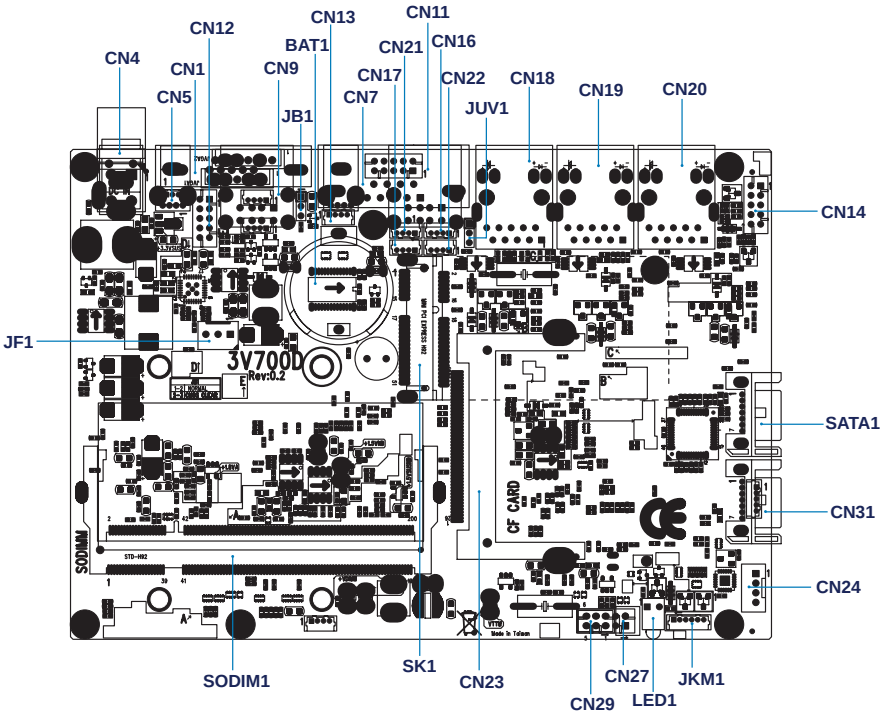
2-3 3V700D-CRXXX Dimension



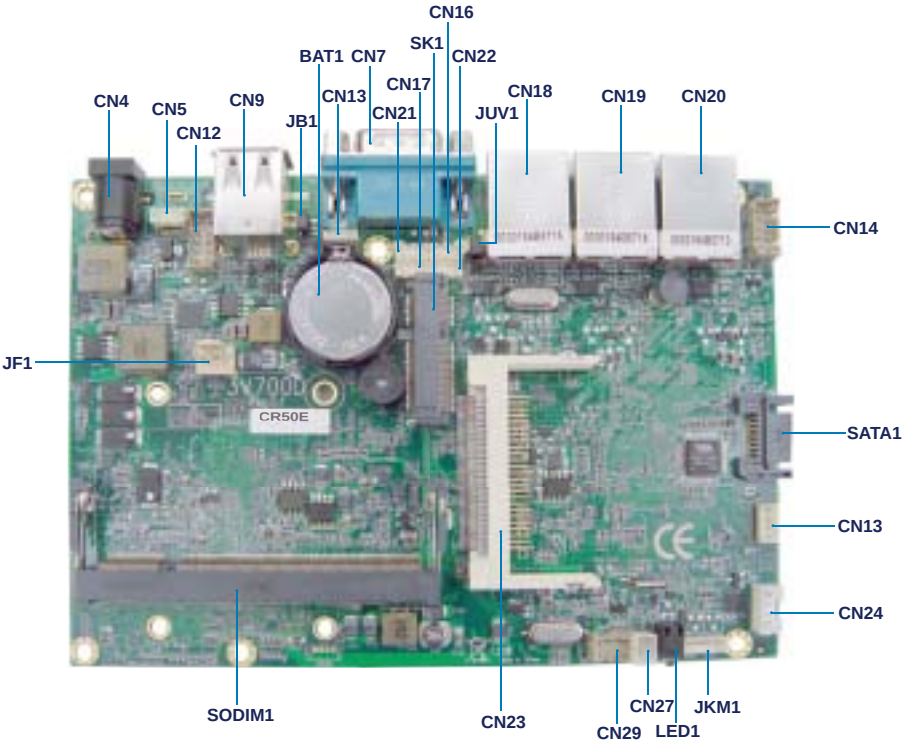
2-3-1 3V700D-VRXXX Dimension



2-4 Layout

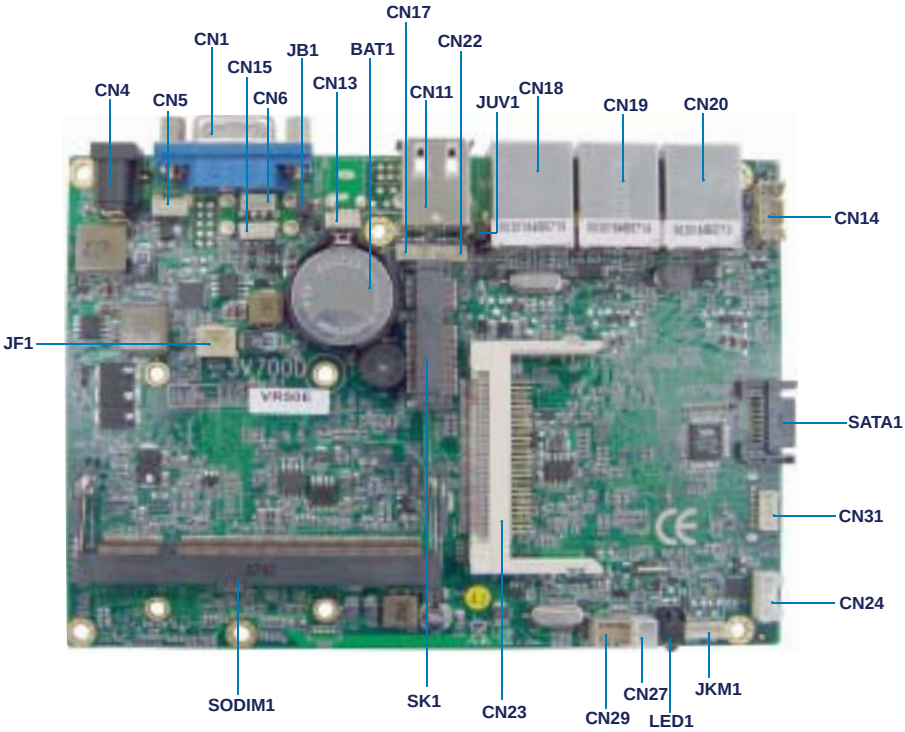

CN4
CN12
CN13
CN11
BAT1
CN21
CN16
CN1
CN9
CN17
CN22
CN18
CN19
CN20
CN5
CN7
JB1
JUV1
CN14
JF1
SATA1
CN31
CN24
SODIM1
SK1
CN23
CN27
JKM1
CN29
LED1
3V700D
Rev:0.2
CF CARD
SODIMM

2-5 3V700D-CRXXX Diagram

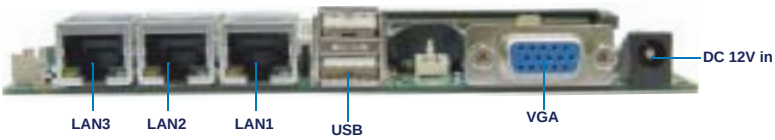


Back Panel

2-5-1 3V700D-VRXXX Diagram



- **Back Panel**



2-6 Install Memory

This motherboard provides one 200-pin Small Outline Dual In-line Memory Module (SODIMM) socket for memory expansion available from minimum memory size of 256MB to maximum memory size of 1GB DDR2 RAM.
DDR2 clock supports: DDRII200, DDRII266
DDR2 Mode supports mixed 64/128/256/512 Mb SDRAM in x8 or x16 configurations.

Valid Memory Configurations

DIMM1	System Accept or Not	Total Memory
		Min. ~ Max.
DS/SS	Accept	256MB~1GB

DS: Double Sided DIMM **SS: Single Sided DIMM**

NOTE!
Make sure the total installed memory does not exceed 1GB; otherwise, the system may hang during startup.

Please refer to page 7 for installation of memory module.

NOTE!

When you install DIMM module fully into the DIMM socket, the eject tab should be locked into the DIMM module very firmly and fit into its indention on both sides.

WARNING!

2. Once you hear " Beep Beep Beep" sounds after turning on the power , please check if the DRAM is installed properly or not.

2-7 List of Jumpers

JB1: CMOS DATA SET

JUV1: USB Voltage Select

2-8 Jumper Setting Description

A jumper is ON as a closed circuit with a plastic cap covering two pins. A jumper is OFF as an open circuit without the plastic cap. Some jumpers have three pins, labeled 1, 2, and 3. You could connect either pin 1 and 2 or 2 and 3.
The below figure 2.2 shows the examples of different jumper settings in this manual.

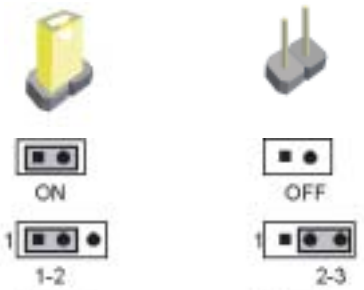


Figure 2.2

All jumpers already have its default setting with the plastic cap inserted as ON, or without the plastic cap as OFF. The default setting may be referred in this manual with a " * " symbol .

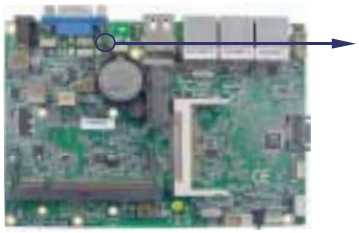
2-9 CMOS Data Set

A battery must be used to retain the motherboard configuration in CMOS RAM. Close pin 1and pin 2 of JB1 to store the CMOS data.

To clear the CMOS, follow the procedures below:
1. Turn off the system and unplug the AC power
2. Remove DC 12V power cable from DC 12V power connector
3. Locate JB1 and close pin 2-3 for a few seconds
4. Return to its normal setting by shorting pin 1-2
5. Connect DC 12V power cable back to DC 12V power connector

Note: Do not clear CMOS unless
1. Troubleshooting
2. Forget password
3. You fail over-clocking system

JB1	Description
***1-2**	**Normal set**
2-3	CMOS data clear



* We use □ to refer to as pin1

2-10 JUV1: USB Voltage Select

JUV1: USB Voltage Select

JB1	Description
***1-2**	**+5V**
2-3	+3.3V

JUV1

+5V*

+3.3V

Chapter-3

Connection

This chapter provides all necessary information of the peripheral's connections, switches and indicators. Always power off the board before you install the peripherals.

3-1 List of Connectors

CN4: DC 12V-IN Power Jack
CN3: DC 12V-IN ATX Power 2x2 pin connector (Share to CN4)
CN1: VGA DB15pin Connector
CN12: VGA wafer connector (Share CN1)
CN7: COM1 DB9pin Connector
CN2: COM1 wafer connector (Share CN7)
CN31: COM2 wafer connector
CN18: LAN1 RJ45 Connector
CN19: LAN2 RJ45 Connector
CN20: LAN3 RJ45 Connector
CN14: Audio wafer connector
SATA1/2: SATA 1/2 SATA connector
CN24: DC +5/+12V output wafer connector
JKM1: PS2 Keyboard /Mouse wafer Connector
JF1: CPU FAN connector
CN25: I2C wafer connector(optional)
CN26: Front Panel wafer connector
CN27: Power button wafer connector (Share CN26)
CN29: LAN LED wafer connector (Share CN26)
CN23: CF 50pin socket
SK1: PCIe Mini Card 52pin socket (Share CN17)

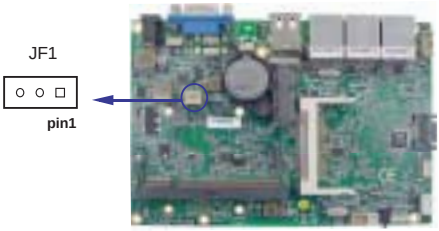
. List of Connectors

CN13: USB port 0 wafer connector
CN17: USB port 1 wafer connector
CN15: USB port 2 wafer connector
CN21: USB port 3 wafer connector
CN6: USB port 4 wafer connector
CN16: USB port 5 wafer connector
CN5: USB port 6 wafer connector
CN22: USB port 7 wafer connector
CN11: USB port 3/5 Type A connector (Share CN16/21)
CN9: USB port 2/4 Type A connector (Share CN6/15)
CN10: USB port 0 Type A connector (Share CN13)
CN8: USB port 6 Type A connector (Share CN5)

3-2 FAN Connector

3V700D provides one CPU fan connector .
CPU Fan Connector- JF1

PIN NO.	Description
1	FAN GND
2	**+12V**
3	FAN speed Sensor

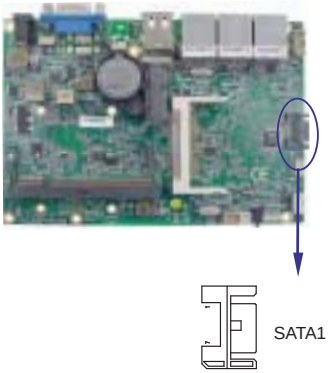


*Note : DC in +12V by switch to FAN power +12V, so DC in need stable +12V input

3-3 Serial ATA

- **SATA1/2: S-ATA IDE 7pin Wafer Connector**

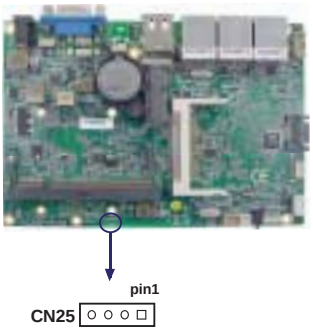
PIN NO.	Description
1	GND
2	DATA TX+
3	DATA TX-
4	GND
5	DATA RX-
6	DATA RX+
7	GND



3-4 I²C Bus Connector(Optional)

CN25: I²C Bus connector 4 pin (1.25mm Wafer connector) (Option)

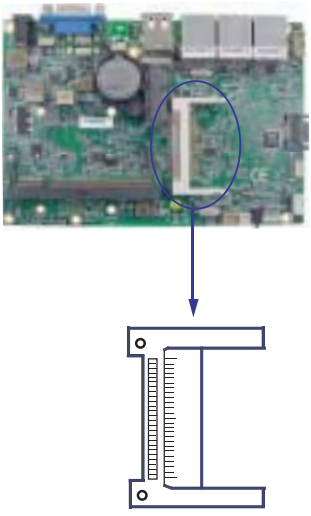
PIN NO.	Description
1	+3.3V or +5V
2	GND
3	I2C CLK
4	I2C DATA



3-5 Compact - Flash Memory Socket

3V700D configures CompactFlash Storage Card in IDE mode.
It will use IDE channel when CompactFlash card is plugged in.
This socket supports CF Card Type I/II socket spec.
CF 50pin Socket ----CN23

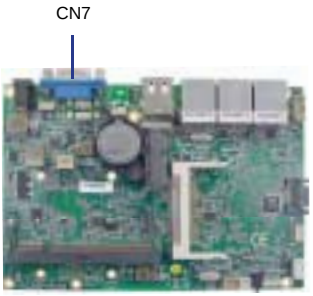
PIN NO.	Description	PIN NO.	Description
1	GND	26	GND(-CD1)
2	DATA3	27	DATA11
3	DATA4	28	DATA12
4	DATA5	29	DATA13
5	DATA6	30	DATA14
6	DATA7	31	DATA15
7	-CS0	32	-CS1
8	GND(A10)	33	GND(-VS1)
9	-ATA_ SEL)	34	-IOR
10	GND(A9)	35	-IOW
11	GND(A8)	36	VCC/-WE(PH)
12	GND(A7)	37	INTR
13	+5V or+3.3V	38	+5V or+3.3V
14	GND(A6)	39	-CSEL
15	GND(A5)	40	NC(-VS2)
16	GND(A4)	41	RESET
17	GND(A3)	42	IORDY
18	SDA2	43	DREQ
19	SDA1	44	-DACK
20	SDA0	45	-DASP
21	DATA0	46	-PDIAG
22	DATA1	47	DATA8
23	DATA2	48	DATA9
24	NC(-IOCS16)	49	DATA10
25	GND(-CD2)	50	GND



3-6 COM Port Connector

CN7: COM1 D-SUB 9pin Connector

PIN NO.	Description	PIN NO.	Description
1	DCD	6	DSR
2	RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	RI
5	GND		



3V700D-CRXXX

CN2: COM1 2X5pin 2.00mm wafer connector

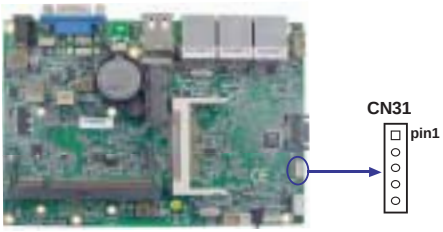
PIN NO.	Description	PIN NO.	Description
1	DCD	2	RXD
3	TXD	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI/VOLTAGE	10	NC

Note: For option share to CN7 connector

CN31: COM2 1X5pin 1.25mm wafer connector (Option)

PIN NO.	Description
1	+5V
2	GND
3	RTS
4	TXD
5	RXD

Note: All signal for TTL level

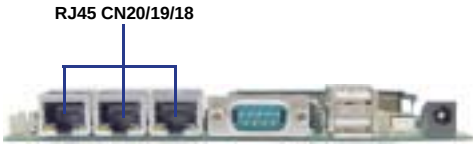


3-7 LAN Port

The Fast Ethernet controller provides 32-bit performance, PCI bus master capability, and full compliance with IEEE 802.3 10/100Based-T specification.
For 10/100Base-T operation, please connect the network connection by plugging one end of the cable into the RJ-45 jack of the CN10 Connector.

CN18/19/20: LAN1/LAN2/LAN3 RJ45 connector

PIN NO.	Description	PIN NO.	Description
1	TD0-/TX+	5	TD2-/NC
2	TD0+/TX-	6	TD2+/RX-
3	TD1-/RX+	7	TD3-/NC
4	TD1+/NC	8	TD3+/NC



LAN Led

RJ45 LAN Connector--- LED define for Intel Giga/100/10MB Connector

Back side connector	Red LED	Green LED	Yellow LED
Indicate	GIGA LAN Link(light)	100Mb LAN Link(light)	10Mb LAN Link(light)
Fron side	LAN LED Header		
Indicate	LAN10/100/1000BT Actived (light)		

RJ45 LAN Connector--- LED define 10/100Mb Connector

LAN LED :

Speed	100Mbps			10Mbps		
Indicate	Back side		Fron side	Back side		Fron side
	Link Led	ACT Led	ACT Led	Link Led	ACT Led	ACT Led
LAN light	Orange	Green	Orange	Orange	Green	Orange

3-8 USB Header

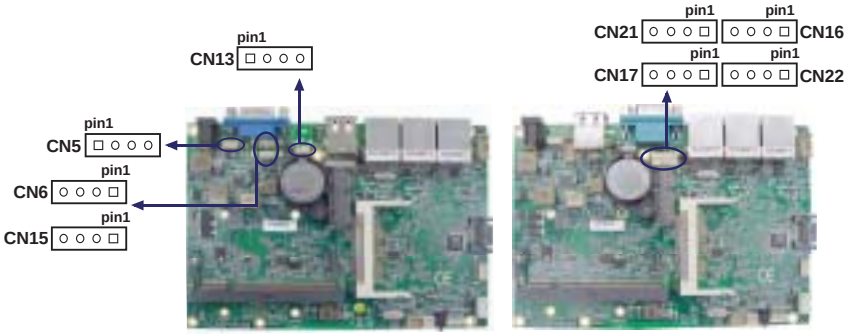
CN5/6/13/15/16/17/21/22: USB 0/1/2/3/4/5/6/7 4pin 1.25mm wafer connector

PIN NO.	Description
1	+5V or+3.3V or+5Vsb
2	USB DATA 0/1-
3	USB DATA 0/1+
4	GND

Note1: CN13/CN16/CN21 USB Voltage For +5V

Note2: CN5/CN6/CN15 USB Voltage For +5VSB

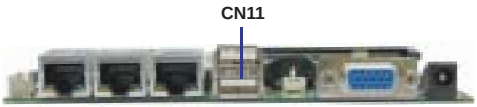
Note3: CN17/CN22 Voltage For JUV1 Jumper Select +5V or +3.3V



3V700D-VR50E 3V700D-CR50E

CN11: USB port 3/5 Type A connector

PIN NO.	Description
1	+5V
2	USB DATA 3/5-
3	USB DATA 3/5
4	GND



Note: For option share to CN16/21 connector

CN9: USB port 2/4 Type A connector

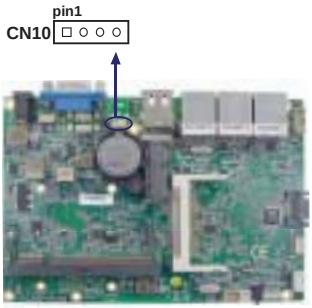
PIN NO.	Description
1	+5VSB
2	USB DATA 2/4-
3	USB DATA 2/4+
4	GND

Note: For option share to CN6/15 connector

CN10: USB port 0 Type A connector

PIN NO.	Description
1	+5V
2	USB DATA 0-
3	USB DATA 0+
4	GND

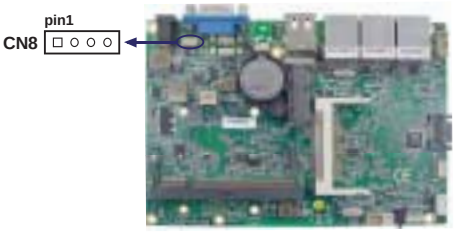
Note: For option share to CN13 connector



CN8: USB port 6 Type A connector

PIN NO.	Description
1	+5VSB
2	USB DATA 6-
3	USB DATA 6+
4	GND

Note: For option share to CN5 connector



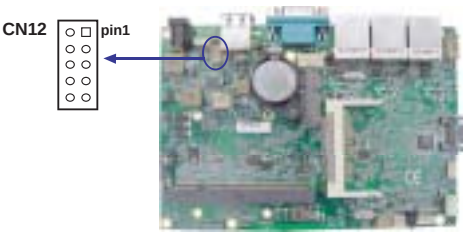
3-9 VGA Port Header

CN1: VGA D-SUB 15pin Connector

PIN NO.	Description	PIN NO.	Description	PIN NO.	Description
1	RED	6	GND	11	NC
2	GREEN	7	GND	12	DDC DATA
3	BULE	8	GND	13	H-SYNC
4	NC	9	NC	14	V-SYNC
5	GND	10	GND	15	DDC CLOCK

CN12: VGA 2x5pin 2.00mm wafer connector

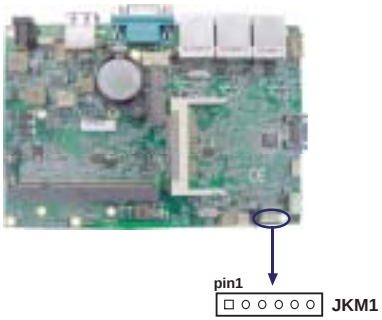
PIN NO.	Description	PIN NO.	Description
1	BULE	2	GND
3	GND	4	DDC CLOCK
5	GREEN	6	V-SYNC
7	GND	8	H-SYNC
9	RED	10	DDC DATA



3-10 PS2 Keyboard /Mouse wafer Connector

JKM1 6PIN 1.25mm Wafer connector (Option)

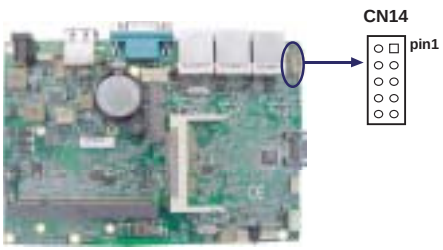
PIN NO.	Description
1	+5V
2	KB DATA
3	KB CLOCK
4	GND
5	MS DATA
6	MS CLOCK



3-11 Audio wafer connector

CN14: Audio 2x5pin 2.00mm wafer Connector (Option)

PIN NO.	Description	PIN NO.	Description
1	Line-out-R	2	MIC-IN
3	Line-in-R	4	GND
5	GND	6	GND
7	Line-in-L	8	NC
9	Line-out-L	10	MIC-IN



3-12 DC 12V-IN Power Jack

CN4: DC 12V-IN external Connector (2pin 2.5mm Jack)

PIN NO.	Description
1.2	+12V DC-IN
3.4	GND

CN3: DC 12V-IN Internal Connector (ATX Power 2x2pin connector)

PIN NO.	Description
3.4	+12V DC-IN
1.2	GND

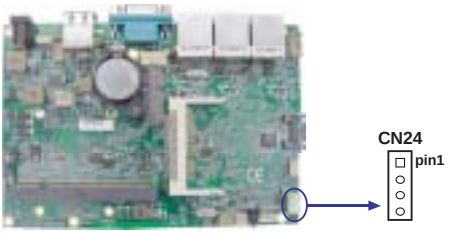
Note: For option share to CN4 connector

3-13 DC +5/+12V output wafer connector

CN24: DC +5/+12V output connector (4pin 2.0mm Wafer)

PIN NO.	1	2	3	4
Description	+5V	GND	GND	+12V *

*Note: DC in +12V by switch to DC-out voltage +12V So DC in need stable +12V input



3-14 Front Panel wafer connector

CN26: Front Panel Function 2x5pin 2.00mm wafer connector

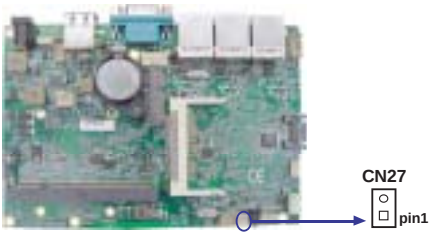
PIN NO.	Description	PIN NO.	Description
1	Power button pin	2	Power button GND
3	Reset pin	4	Reset GND
5	Power LED -	6	Power LED +
7	HDD LED-	8	HDD LED+
9	LAN LED-	10	LAN LED+

Note: For option share to CN27 connector

3-15 Power button wafer connector

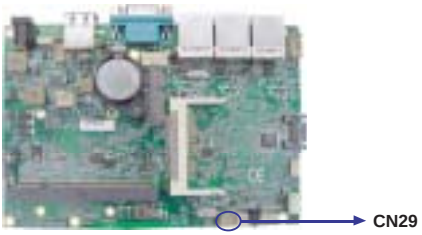
CN27: Power button 2pin 2.00mm wafer connector (Share CN26)

PIN NO.	Description
1	Power button pin
2	Power button pin GND



3-16 LAN LED wafer connector

CN29: LAN LED wafer connector (Share CN26)



3-17 For F75111N I2C watch dog timer device:

DIO function: (Hardware no pin out)
Logic 0 Level :+0.5V Max , Logic 1 Level : +4V Min
WatchDog function:
The system will be issued reset. When WDT is enable the hardware start down counter to zero. The reset timer have 10~20% tolerance upon the Temperature.
Note: Please refer to “Appendix A” for sample code or “Menu” for detail description

----
Sample W75IO Demo Program
----
1. DI
2. DO
3. WatchDog
4. Exit

Please select demo function<1-4>?
1. DI: Input DI value (0~FF, DI bit 0~7 HI level)
2. DO: Input DO value (0~FF, DO bit 0~7 HI level)
3. WatchDog: Input WatchDog Value (0~256 sec)
4. Exit: Exit program

3-18 I2C WDT/DIO W75IO.h Reference sample code:

Compile platform:TC30 & DOS 6.22

```
// Sample.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "conio.h"
#include <string.h>
#include <stdlib.h>

#include <iostream>

#include "W75IO.h"              /* include W75IO.h    */

#define DEMO_DI       1
#define DEMO_DO       2
#define DEMO_WDT      3
#define DEMO_EXIT     4

/* demo function */

int         menu();
BOOL        W75IO_init();
void        W75IO_DO();
void        W75IO_DI();
void        W75IO_WDT();

int   stoi(char* str)
{
     if (( strlen(str) < 0 ) || ( strlen(str) >= 3 ))
          return -1;

     int value = 0;

     for (int i=0; i<strlen(str) ;i=i+1)
     {
          int result=0;

          switch (str[i])
          {
               case '0':result=0;
               break;

               case '1':result=1;break;
               case '2':result=2;break;
               case '3':result=3;break;
               case '4':result=4;break;
               case '5':result=5;break;
               case '6':result=5;break;
               case '7':result=7;break;
               case '8':result=8;break;
               case '9':result=9;break;
               case 'a':
               case 'A':result=10;break;
               case 'b':
               case 'B':result=11;break;
               case 'c':
               case 'C':result=12;break;
               case 'd':
               case 'D':result=13;break;
               case 'e':
               case 'E':result=14;break;
               case 'f':
               case 'F':result=15;break;
```

```
                default:
                    result=-1;
                break;
            }

            if (result == -1)
                return -1;

            if (strlen(str) == 1 )
                value=value+result;

            if (strlen(str) == 2 )
            {
                if ( i == 0 )
                {
                    value=value+result*16;
                }
                if ( i == 1 )
                {
                    value=value+result;
                }
            }
        }

        return value;
}

BOOL   W75IO_init()
{
        /* First step :to init W75IO.DLL */
        if (InitializeW75IO())
        {
            /* init onboard W75IO chipset */
            if (InitInternalW75IO())
                return TRUE;
            else
            {
                printf("ERROR:init Onboard W75IO chipset!!!!.\n");
                return FALSE;
            }
        }
        else
        {
            printf("ERROR:Can't init W75IO.DLL!!!.\n");
            return FALSE;
        }

        return FALSE;
}

void W75IO_DO()
{
        int value;

        printf("Please enter your want output value (0-255)   \n");
        cin>>value;

        /* use onboard W75IO output value*/
        InterDigitalOutput(value);
}

void W75IO_DI()
{
        /* read & show onboard W75IO input value*/
```

```
printf("read input input value %d\n",InterDigitalInput());
    getch();
}

void W75IO_WDT()
{
    /* get WDT timer value*/
    int value;

    printf("Please enter WatchDog Timer value (0-255)\n");


    cin>>value;

    /* check value */
    int timer=value;

    if (timer > 255)
    {
        printf("must be > 255\n");

        getch();

        return;
    }


    /* sete onboard W75IO use secord as WatchDog Timer unit */
    SetInterWDTUnit(FALSE);

    /* Enable onboard W75IO WatchDog Timer   */
    EnableInterWDT(timer);


    /* show message... */
    for (int i=0;i<timer;i++)
    {
        printf("the System will reboot after %3d sec ....\n",timer-i);
        Sleep(1000);
    }
}

int menu()
{
    int iOption;

    /* First clean screen */
    system("cls");

    /* show demo menu()*/
    printf("\n");
    printf("---------------------------------- \n");
    printf("Sample W75IO Demo Program \n");
    printf("----------------------------------\n");
    printf("1.DI\n");
    printf("2.DO\n");
    printf("3.WatchDog\n");
    printf("4.Exit\n");
    printf("---------------------------------- \n");
    printf("Please select demo function(1-4)?");

    /* get user select */
    cin>>iOption;

    return iOption;
```

```
}

int main(int argc, char* argv[])
{


     /* get user enter value*/


     if (W75IO_init())
     {
          while (1)
          {
               int option =menu();

               if ( option == 4 )
                    break;

               switch (option)
               {
                    case DEMO_DI:
                         W75IO_DI();
                    break;
                    case DEMO_DO:
                         W75IO_DO();
                    break;

                    case DEMO_WDT:
                         W75IO_WDT();
                    break;

                    case DEMO_EXIT:
                    break;
               }
          }
     }
     return 0;
}
```

3-18-1 I2C WDT/DIO F75111N reference sample code:

Compile platform: Windows (include SMBus.h)

Contents

1. Introduction
   1.1 Initial Internal F75111 port address (0x9c)
   1.2 Set F75111 DI/DO (sample code as below Get Input value/Set output value)
   1.3 Enable/Disable WDT
   1.4 PULSE mode
2. Initial internal F75111
3. Set output value
4. Get input value
5. Enable Watch Dog
6. Disable Watch Dog
7. Define F75111 pin

1. Introduction

1-1. Initial Internal F75111 port address (0x9c)

define GPIO1X, GPIO2X, GPIO3X to input or output and Enable WDT function pin

1-2. Set F75111 DI/DO (sample code as below Get Input value/Set output value)

DO: InterDigitalOutput(BYTE byteValue))

DI: InterDigitalInput()

1-3. Enable/Disable WDT

Enable : F75111_SetWDTEnable (BYTE byteTimer)

Disable: F75111_SetWDTDisable ()

1-4. PULSE mode

Sample to setting GP33, 32, 31, 30 output 1mS low pulse signal.

```
{
//This is setting low pulse output
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_CONTROL,      0x00);
//This selects the pulse width to 1mS
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_PULSE_WIDTH_CONTROL, 0x01);
//This is setting the GP33, 32, 31, 30 to output function.
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_CONTROL_MODE,      0x0F);
//This is setting the GP33, 32, 31, 30
    this->Write_Byte(F75111_INTERNAL_ADDR, GPIO3X_Output_Data ,      0x0F);
}
```

2. Initial internal F75111

```
void F75111::InitInternalF75111()
    {
     //set GPIO1X to Input  function
            this->Write_Byte(F75111_INTERNAL_ADDR,GPIO1X_CONTROL_MODE  ,0x00);
     //set GPIO3X to Input  function
            this->Write_Byte(F75111_INTERNAL_ADDR,GPIO3X_CONTROL_MODE  ,0x00);
     //set GPIO2X to Output function
           this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_CONTROL_MODE  ,0xFF);
    //Enable WDT OUT function
          this->Write_Byte(F75111_INTERNAL_ADDR,F75111_CONFIGURATION, 0x03);
    }
```

3. Set output value

```
void F75111::InterDigitalOutput(BYTE byteValue)
{
       BYTE byteData = 0;
       byteData = (byteData & 0x01 )? byteValue + 0x01 : byteValue;
       byteData = (byteData & 0x02 )? byteValue + 0x02 : byteValue;
       byteData = (byteData & 0x04 )? byteValue + 0x04 : byteValue;
       byteData = (byteData & 0x80 )? byteValue + 0x08 : byteValue;
       byteData = (byteData & 0x40 )? byteValue + 0x10 : byteValue;
       byteData = (byteData & 0x20 )? byteValue + 0x20 : byteValue;
       byteData = (byteData & 0x10 )? byteValue + 0x40 : byteValue;
       byteData = (byteData & 0x08 )? byteValue + 0x80 : byteValue;          // get value bit by bit
// write byteData value via GPIO2X output pin
     this->Write_Byte(F75111_INTERNAL_ADDR,GPIO2X_OUTPUT_DATA,byteData);
}
```

4. Get Input value

```
BYTE F75111::InterDigitalInput()
{
       BYTE byteGPIO1X = 0;
       BYTE byteGPIO3X = 0;
       BYTE byteData   = 0;
  // Get value from GPIO1X
        this->Read_Byte(F75111_INTERNAL_ADDR,GPIO1X_INPUT_DATA,&byteGPIO1X) ;
  // Get value from GPIO3X
        this->Read_Byte(F75111_INTERNAL_ADDR,GPIO3X_INPUT_DATA,&byteGPIO3X) ;
```

```
    byteGPIO1X = byteGPIO1X  & 0xF0;                // Mask unuseful value
    byteGPIO3X = byteGPIO3X  & 0x0F;                // Mask unuseful value

    byteData = ( byteGPIO1X & 0x10 )? byteData + 0x01 : byteData;
    byteData = ( byteGPIO1X & 0x80 )? byteData + 0x02 : byteData;
    byteData = ( byteGPIO1X & 0x40 )? byteData + 0x04 : byteData;
    byteData = ( byteGPIO3X & 0x01 )? byteData + 0x08 : byteData;

    byteData = ( byteGPIO3X & 0x02 )? byteData + 0x10 : byteData;
    byteData = ( byteGPIO3X & 0x04 )? byteData + 0x20 : byteData;
    byteData = ( byteGPIO3X & 0x08 )? byteData + 0x40 : byteData;
    byteData = ( byteGPIO1X & 0x20 )? byteData + 0x80 : byteData;
// Get correct DI value from GPIO1X & GPIO3X

      return byteData;
}
```

5. Enable WatchDog

```
void F75111_SetWDTEnable (BYTE byteTimer)
{
// set WatchDog range and timer
     WriteByte(F75111_INTERNAL_ADDR,WDT_TIMER_RANGE  ,byteTimer);
// Enable WatchDog, Setting WatchDog configure
     WriteByte(F75111_INTERNAL_ADDR,WDT_CONFIGURATION,WDT_TIMEOUT_FLAG | WDT_ENABLE |
     WDT_PULSE | WDT_PSWIDTH_100MS);
}
```

6. Disable WatchDog

```
void F75111_SetWDTDisable ()
{
      WriteByte(F75111_INTERNAL_ADDR,WDT_CONFIGURATION,0x00);
}
```

7. Define F75111 pin in F75111.h

```
//-----------------------------------------------------------------------------------------------
#define   F75111_INTERNAL_ADDR                  0x9C  //  OnBoard  F75111 Chipset
#define   F75111_EXTERNAL_ADDR                  0x6E  //  External F75111 Chipset
//-----------------------------------------------------------------------------------------------
#define   F75111_CONFIGURATION                  0x03  //  Configure GPIO13 to WDT2 Function
//-----------------------------------------------------------------------------------------------
#define   GPIO1X_CONTROL_MODE                   0x10  //  Select Output Mode or Input Mode
#define   GPIO2X_CONTROL_MODE                   0x20  //  Select GPIO2X Output Mode or Input Mode
#define   GPIO3X_CONTROL_MODE                   0x40  //  Select GPIO3X Output Mode or Input Mode
//-----------------------------------------------------------------------------------------------
#define   GPIO1X_INPUT_DATA                     0x12  //  GPIO1X Input
#define   GPIO3X_INPUT_DATA                     0x42  //  GPIO3X Input
//-----------------------------------------------------------------------------------------------
#define   GPIO2X_OUTPUT_DATA                    0x21  //  GPIO2X Output
//-----------------------------------------------------------------------------------------------
#define   GPIO1X_PULSE_CONTROL                  0x13  //  GPIO1x Level/Pulse Control Register
                                                //  0:Level Mode
                                                //  1:Pulse Mode
#define   GPIO1X_PULSE_WIDTH_CONTROL            0x14  //  GPIO1x Pulse Width Control Register
#define   GP1_PSWIDTH_500US                     0x00  //  When select Pulse mode:500us.
#define   GP1_PSWIDTH_1MS                       0x01  //  When select Pulse mode:1ms.
#define   GP1_PSWIDTH_20MS                      0x02  //  When select Pulse mode:20ms.
#define   GP1_PSWIDTH_100MS                     0x03  //  When select Pulse mode:100ms.
//-----------------------------------------------------------------------------------------------
#define   GPIO2X_PULSE_CONTROL                  0x23  //  GPIO2x Level/Pulse Control Register
                                                //  0:Level Mode
                                                //  1:Pulse Mode
#define   GPIO2X_PULSE_WIDTH_CONTROL            0x24  //  GPIO2x Pulse Width Control Register
#define   GP2_PSWIDTH_500US                     0x00  //  When select Pulse mode:500us.
#define   GP2_PSWIDTH_1MS                       0x01  //  When select Pulse mode:1ms.
#define   GP2_PSWIDTH_20MS                      0x02  //  When select Pulse mode:20ms.
#define   GP2_PSWIDTH_100MS                     0x03  //  When select Pulse mode:100ms.
//-----------------------------------------------------------------------------------------------
#define   GPIO3X_PULSE_CONTROL                  0x43  //  GPIO3x Level/Pulse Control Register
                                                //  0:Level Mode
                                                //  1:Pulse Mode
#define   GPIO3X_Output_Data                    0x41  //  GPIO3x Output Data Register
#define   GPIO3X_PULSE_WIDTH_CONTROL            0x44  //  GPIO3x Pulse Width Control Register
#define   GP3_PSWIDTH_500US                     0x00  //  When select Pulse mode:500us.
#define   GP3_PSWIDTH_1MS                       0x01  //  When select Pulse mode:1ms.
```

```
#define    GP3_PSWIDTH_1MS                0x01  //  When select Pulse mode: 1    ms.
#define    GP3_PSWIDTH_20MS               0x02  //  When select Pulse mode: 20  ms.
#define    GP3_PSWIDTH_100MS              0x03  //  When select Pulse mode:100 ms.
//-------------------------------------------------------------------------------------------------
#define    WDT_TIMER_RANGE                 0x37  //  0-255 (secord or minute program by
                                                 //   WDT_UNIT)
#define    WDT_CONFIGURATION              0x36  //  Configure WDT Function
#define    WDT_TIMEOUT_FLAG               0x40  //  When watchdog timeout.this bit will be set
to 1.
#define    WDT_ENABLE                     0x20  //  Enable watchdog timer
#define    WDT_PULSE                      0x10  //  Configure WDT output mode
                                                 //  0:Level Mode
                                                 //  1:Pulse Mode

#define    WDT_UNIT                       0x08  //  Watchdog unit select.
                                                 //  0:Select second.
                                                 //  1:Select minute.

#define    WDT_LEVEL                      0x04  //  When select level output mode:
                                                 //  0:Level low
                                                 //  1:Level high

#define    WDT_PSWIDTH_1MS                0x00  //  When select Pulse mode:  1     ms.
#define    WDT_PSWIDTH_20MS               0x01  //  When select Pulse mode:  20   ms.
#define    WDT_PSWIDTH_100MS              0x02  //  When select Pulse mode:  100  ms.
#define    WDT_PSWIDTH_4000MS             0x03  //  When select Pulse mode:  4     s.
```

Contents

1. Write_Byte Mode
2. Read_Byte Mode
3. Check Device (F75111)
4. SMBus_Clear
5 SMBus_Wait
6. SMBus_Busy
7. IO_Write
8. IO_Read
9. Define SMBus IO address
10. Define SMBus pin in SMBus.

1. Write Byte Mode

```
int SMBus::Write_Byte(WORD dwSlave, BYTE pCmd, BYTE pByte)
{
      this->SMBus_Clear();                              // Clear SMBus data first

if (this->SMBus_Busy())                                 // Check SMBus busy or not, return busy if busy
          return SMBUS_BUSY;

    this->IO_Write(SMBHSTADD  , dwSlave & ~1 );         // write address in first variable
    this->IO_Write(SMBHSTCMD  , pCmd );                 // write command in second variable
    this->IO_Write(SMBHSTDAT0 , pByte );                // write data in third variable
    this->IO_Write(SMBHSTCNT  , SMBHSTCNT_START | SMBHSTCNT_BYTE);
    // Sent start command to SMBus control register

return (int)this->SMBus_Wait();                         // return wait command when SMBus finish the job
}
```

2. Read Byte Mode

```
int SMBus::Read_Byte(WORD dwSlave, BYTE pCmd, BYTE *pByte)
{
     this->SMBus_Clear();                               // Clear SMBus data first

     if (this->SMBus_Busy())                            // Check SMBus busy or not, return busy if busy
      return SMBUS_BUSY;

      this->IO_Write(SMBHSTADD  , dwSlave | 1 );        // write address in first variable
      this->IO_Write(SMBHSTCMD  , pCmd );               // write command in second variable
      this->IO_Write(SMBHSTCNT  , SMBHSTCNT_START | SMBHSTCNT_BYTE);
// Sent start command to SMBus control register
```

```
    int ret = this->SMBus_Wait();                          // Check SMBus Status

    if (ret == SMBUS_OK)                                   // If SMBus Stand by
    {
        *pByte = (BYTE)this->IO_Read(SMBHSTDAT0) &0xFF;  // Get SMBus host data value
    }
    return ret;                                            // reture SMBus status
}
```

3. Check Device (F75111)

```
BOOL SMBus::CheckDevice(WORD wDeviceAddress)
{
    int ret;
    this->SMBus_Clear();                                   // Clear SMBus data first

    if (this->SMBus_Busy())                                // Check SMBus busy or not, return busy if busy
        return SMBUS_BUSY;

    this->IO_Write(SMBHSTADD , wDeviceAddress & ~1 );   // write address in first variable
    this->IO_Write(SMBHSTCNT , SMBHSTCNT_START | SMBHSTCNT_SENDRECV);
// Sent start command to SMBus control register

    ret = this->SMBus_Wait();                          // Check SMBus Status

    if (ret == SMBUS_OK)                               // Check device exist or not, if exist return ture else false
        return TRUE;
    else
        return FALSE;
}
```

4. SMBus_Clear

```
void SMBus::SMBus_Clear()
{
    this->IO_Write(SMBHSTSTS ,0xFF);                   // Clear SMBus status
    this->IO_Write(SMBHSTDAT0,0x0 );                   // Clear SMBus data
}
```

5. SMBus_Wait

```
int SMBus::SMBus_Wait()
{
    int          timeout = SMBUS_TIMEOUT;
    DWORD  dwValue;

    while (timeout--)
    {
        Sleep(10);                                          // I/O Delay
        dwValue = IO_Read(SMBHSTSTS) & 0x00FF;              // Read Host Status Register
        if( dwValue & SMBHSTSTS_INTR )                      // if status value equal
    SMBHSTSTS_INTR, return SMBus_OK
        {
            return SMBUS_OK;
        }
    // if status value equal SMBHSTSTS_FAILED, return SMBHSTSTS_FAILED
        if( dwValue & SMBHSTSTS_FAILED )
        {
            printf("SMBus Action FAILED! %x\n",dwValue);
            return SMBHSTSTS_FAILED;
        }
    // if status value equal SMBHSTSTS_COLLISION, return SMBHSTSTS_COLLISION
        if(dwValue & SMBHSTSTS_COLLISION)
        {
            printf("SMBus Action COLLISION! %x\n",dwValue);
            return SMBHSTSTS_COLLISION;
        }
    // if status value equal SMBHSTSTS_ERROR, return SMBHSTSTS_ERROR
        if(dwValue & SMBHSTSTS_ERROR)
        {
            printf("SMBus Action ERROR! %x\n",dwValue);
            return SMBHSTSTS_ERROR;
        }
    }
    return SMBUS_BUSY;
}
```

6. SMBus_Busy

```
BOOL SMBus::SMBus_Busy()
{
// Check SMBus status if equal SMBHSTSTS_BUSY
    if( (this->IO_Read(SMBHSTSTS) & SMBHSTSTS_BUSY ) == 1 )
        return TRUE;                        // return true
    else
        return FALSE;                       // else retrun false
}
```

7. IO_Write

```
void SMBus::IO_Write(WORD dwOffset, BYTE dwData)
{
// Set dwData value to assigned address
    SetPortVal(this->m_MapIOAddress+dwOffset, dwData,1);
}
```

8. IO_Read

```
BYTE SMBus::IO_Read(WORD dwOffset)
{
    DWORD dwAddrVal;

// Get dwAddrVal value from assigned address
    GetPortVal(this->m_MapIOAddress+dwOffset,&dwAddrVal,1);
    return (BYTE)(dwAddrVal & 0x0FF);
}
```

9. Define SMBus IO address

```
SMBus::SMBus()
{
  this->m_MapIOAddress = 0x500;
}
```

10. Define SMBus pin in SMBus.h

```
#define    SMBHSTSTS                              0x00    // SMBus Host  Status Register Offset
#define    SMBHSTSTS_BUSY                 0x01    // SMBus Host -> 0000-0001 Busy
#define    SMBHSTSTS_INTR                 0x02    // SMBus Host -> 0000-0010 Interrupt / complection
#define    SMBHSTSTS_ERROR                0x04    // SMBus Host -> 0000-0100 Error
#define    SMBHSTSTS_COLLISION            0x08    // SMBus Host -> 0000-1000 Collistion
#define    SMBHSTSTS_FAILED               0x10    // SMBus Host -> 0001-0000 Failed
//-------------------------------------------------------------------------------
#define    SMBHSTCNT                      0x02    // SMBus Host Contorl Register Offset
#define    SMBHSTCNT_KILL                 0x02    // SMBus Host Contorl -> 0000 0010 Kill
#define    SMBHSTCNT_QUICK                0x00    // SMBus Host Contorl -> 0000 0000 quick (default)
#define    SMBHSTCNT_SENDRECV             0x04    // SMBus Host Contorl -> 0000 0100 Byte
#define    SMBHSTCNT_BYTE                 0x08    // SMBus Host Contorl -> 0000 1000 Byte Data
#define    SMBHSTCNT_WORD                 0x0c    // SMBus Host Contorl -> 0000 1100 Word Data
#define    SMBHSTCNT_BLOCK                0x14    // SMBus Host Contorl -> 0001 0100 Block
#define    SMBHSTCNT_START                0x40    // SMBus Host Contorl -> 0100 0000 Start
//-------------------------------------------------------------------------------
#define    SMBHSTCMD                      0x03    // SMBus Host Command    Register Offset
#define    SMBHSTADD                      0x04    // SMBus Host Address    Register Offset
#define    SMBHSTDAT0                     0x05    // SMBus Host Data0      Register Offset
#define    SMBHSTDAT1                     0x06    // SMBus Host Data1      Register Offset
#define    SMBBLKDAT                      0x07    // SMBus Host Block Data Register Offset
//-------------------------------------------------------------------------------
//     SMBus Bus Status Code
//-------------------------------------------------------------------------------
#define    SMBUS_OK                       0x0     // SMBUS OK
#define    SMBUS_BUSY                     0x1     // SMBUS BUSY
#define    SMBUS_INT                      0x2     // SMBUS INTR
#define    SMBUS_ERROR                    0x4     // SMBUS ERROR
//-------------------------------------------------------------------------------
#define    SMBUS_TIMEOUT                  100
```

Chapter 4

Introduction of BIOS

The BIOS is a program located in the Flash Memory on the motherboard.
This program is a bridge between motherboard and operating system.
When you start the computer, the BIOS program gains control. The BIOS first operates an auto-diagnostic test called POST (Power on Self Test) for all the necessary hardware, it detects the entire hardware devices and configures the parameters of the hardware synchronization. After these tasks are completed, BIOS will give control of the computer back to operating system (OS). Since the BIOS is the only channel for hardware and software to communicate with, it is the key factor of system stability and of ensuring your system performance at best.

In the BIOS Setup main menu of Figure 3-1, you can see several options. We will explain these options in the following pages. First, let us see the function keys you may use here:

Press <Esc> to quit the BIOS Setup.
Press ↑↓← → (up, down, left, right) to choose the option you want to confirm or modify.
Press <F10> to save these parameters and to exit the BIOS Setup menu after you complet the setup of BIOS parameters.
Press Page Up/Page Down or +/- keys to modify the BIOS parameters for the active option.

4-1 Enter Setup

Power on the computer and press <Del> immediately to enter Setup.
If the message disappears before your respond but you still wish to enter Setup, restart the system by turning it OFF then ON button on the system case. You may also restart the system by simultaneously pressing <Ctrl>, <Alt> and <Delete> keys. If you do not press the keys at the proper time and the system does not boot, an error message will display and you will be asked to

Press <F1> to continue, <Ctrl-Alt-Esc> or <Del> to enter Setup

4-2 Getting Help

Main Menu

The on-line description of the highlighted setup function is displayed at the bottom of the screen.

Status Page Setup Menu/ Option Page Setup Menu

Press F1 to pop up a small help window that describes the appropriate keys to use and the possible selections for the highlighted item.
To exit the Help Window, press <Esc>.

4-3 The Main Menu

Once you enter Award BIOS CMOS Setup Utility, the Main Menu will appear on the screen. The Main Menu has fourteen setup functions and two exit choices.
Use arrow keys to select among these items. Press <Enter> to accept or enter the sub-menu.

Phoenix-AwardBIOS CMOS Setup Utility

▶ Standard CMOS Features	▶ Frequency/Voltage Control
▶ Advanced BIOS Features	Load Fail-Safe Defaults
▶ Advanced Chipset Features	Load Optimized Defaults
▶ Integrated Peripherals	Set Supervisor Password
▶ Power Management Setup	Set User Password
▶ PnP/PCI Configurations	Save & Exit Setup
▶ PC Health Status	Exit Without Saving

Esc : Quit F9 : Menu in BIOS ↑↓←→ : Select Item
F10 : Save & Exit Setup

Time, Date, Hard Disk Type…

Standard CMOS Features
This Menu is for basic system configurations.

Advanced BIOS Features
This menu is to set the Advanced Features available in your system.

Advanced Chipset Features
This menu is to change the values in the chipset registers and optimize your system performance.

Integrated Peripherals
This menu is to specify your settings for integrated peripherals.

Power Management Setup
This menu is to specify your settings for power management.

PnP/PCI configurations
This entry appears if your system supports PnP/PCI.

PC Health Status
This entry shows your PC health status.

Frequency/Voltage Control
This menu is to specify your settings for Miscellaneous Control.

Load Optimized Defaults
Use this menu to load the BIOS default values that are factory settings for optimal system operation performance. It is provided for the skillful users who want to push the motherboard to get better performance. The Optimized setting does not go through all the reliability and compatibility tests, it is verified only with limited configurations and loading (for example, a system that is equipped with only one VGA card and one DIMM. Do not use Optimized setting unless you fully understand the items in chipset setup menu.

Load Standard Defaults
Use this menu to load the BIOS default values for the optimum system performance. Standard Defaults settings are relatively more reliable than Optimized Defaults. All the product verifications, compatibility test reports and manufacture quality controls are based on "Standard Defaults". We suggest you to use this setting for normal operation. "Standard Defaults" is not the slowest setting for this motherboard. If you need to verify an unstable problem, you may manually set the parameter in "Advanced Chipset Features" to get slower and safer setting.

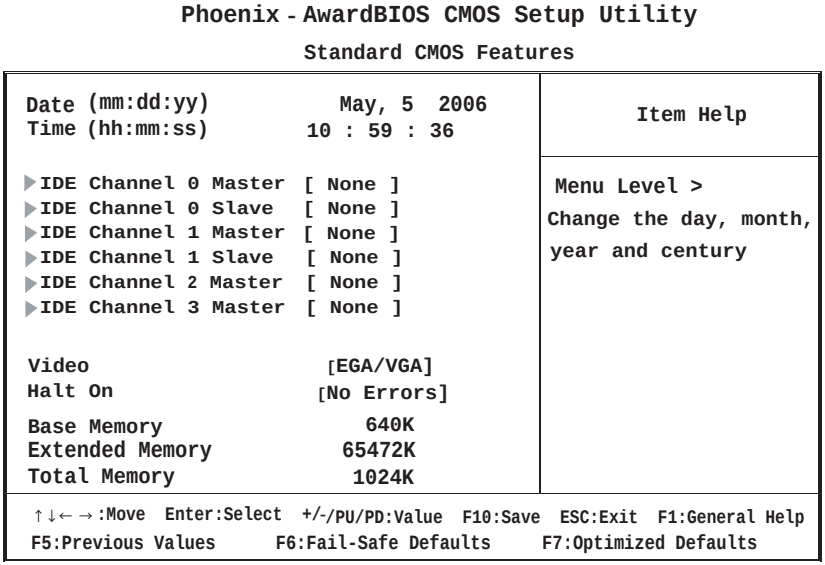
Set Supervisor/User Password
This menu is to set User and Supervisor Passwords.

Save & Exit Setup
Save CMOS values modified to CMOS and exit setup.

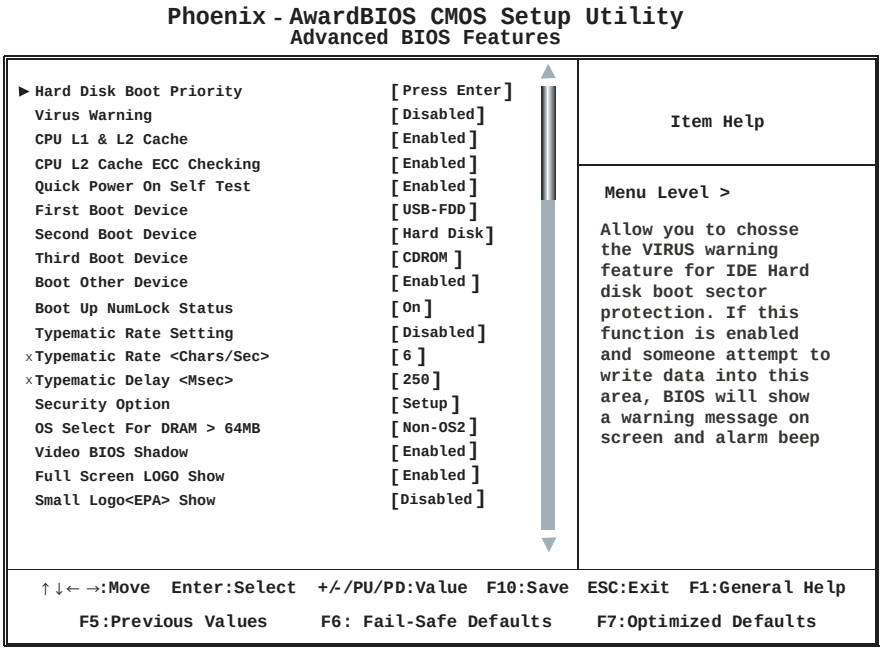
Exit Without Saving
Abandon all the CMOS values modified and exit setup.

4-4 Standard CMOS Features

The items in Standard CMOS Setup Menu are divided into several categories. Each category includes no, one or more than one setup items. Use the arrow keys to highlight the item and then use the <PgUp> or <PgDn> keys to select the value you want to modify with this item.


Phoenix - AwardBIOS CMOS Setup Utility
Standard CMOS Features
Date (mm:dd:yy) May, 5 2006
Time (hh:mm:ss) 10 : 59 : 36
IDE Channel 0 Master [None]
IDE Channel 0 Slave [None]
IDE Channel 1 Master [None]
IDE Channel 1 Slave [None]
IDE Channel 2 Master [None]
IDE Channel 3 Master [None]
Video [EGA/VGA]
Halt On [No Errors]
Base Memory 640K
Extended Memory 65472K
Total Memory 1024K
Item Help
Menu Level >
Change the day, month, year and century
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

4-5 Advanced BIOS Features


Phoenix - AwardBIOS CMOS Setup Utility
Advanced BIOS Features
▶ Hard Disk Boot Priority [Press Enter]
Virus Warning [Disabled]
CPU L1 & L2 Cache [Enabled]
CPU L2 Cache ECC Checking [Enabled]
Quick Power On Self Test [Enabled]
First Boot Device [USB-FDD]
Second Boot Device [Hard Disk]
Third Boot Device [CDROM]
Boot Other Device [Enabled]
Boot Up NumLock Status [On]
Typematic Rate Setting [Disabled]
x Typematic Rate <Chars/Sec> [6]
x Typematic Delay <Msec> [250]
Security Option [Setup]
OS Select For DRAM > 64MB [Non-OS2]
Video BIOS Shadow [Enabled]
Full Screen LOGO Show [Enabled]
Small Logo<EPA> Show [Disabled]
Item Help
Menu Level >
Allow you to chosse the VIRUS warning feature for IDE Hard disk boot sector protection. If this function is enabled and someone attempt to write data into this area, BIOS will show a warning message on screen and alarm beep
↑↓← →:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6: Fail-Safe Defaults F7:Optimized Defaults

Virus Warning

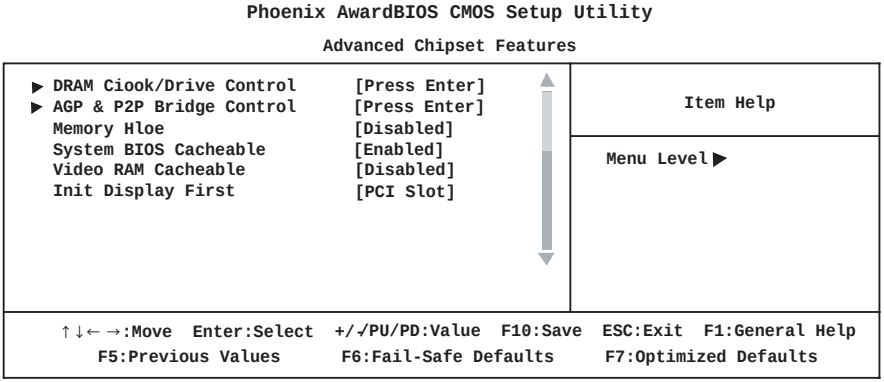
The VIRUS Warning feature can help you protect IDE Hard Disk boot sector.
If this function is enabled, BIOS will show a warning message on screen and alarm beep when somenoe attempts to write data into this area without permission.

Disabled **(**default) No warning message appears when anything attempts to access the boot sector or hard disk partition table.

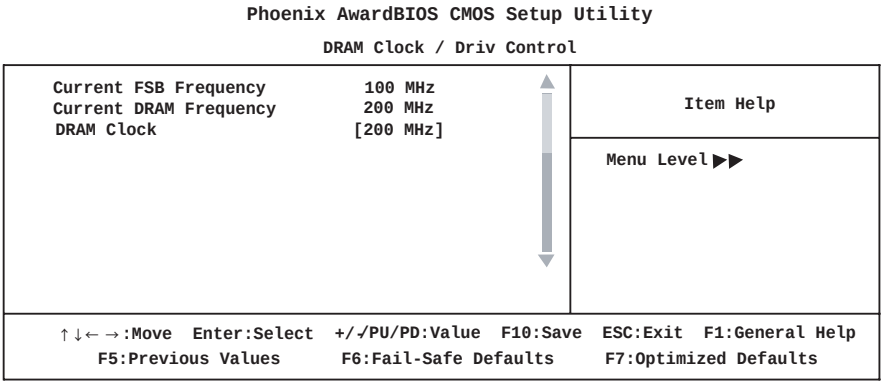
Enabled Activate automatically when the system boots up. The system will show the warning message if anything attempts to access the boot sector of hard disk partition table.

4-6 Advanced Chipset Features

The Advanced Chipset Features Setup option is used to change the values of the chipset registers. These registers control most of the system options in the computer

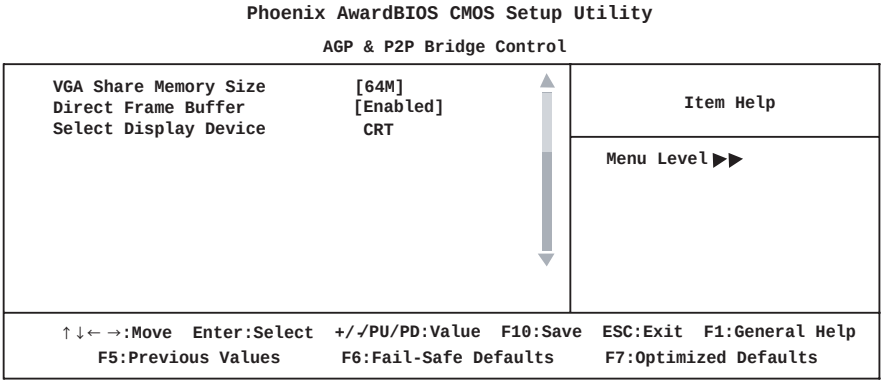

Phoenix AwardBIOS CMOS Setup Utility
Advanced Chipset Features
▶ DRAM Ciook/Drive Control [Press Enter]
▶ AGP & P2P Bridge Control [Press Enter]
Memory Hloe [Disabled]
System BIOS Cacheable [Enabled]
Video RAM Cacheable [Disabled]
Init Display First [PCI Slot]
Item Help
Menu Level ▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

4-6-1 DRAM Clock/Drive Control

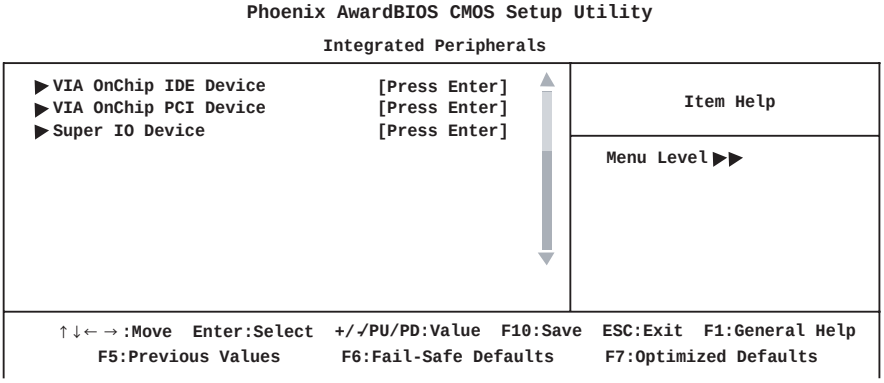

Phoenix AwardBIOS CMOS Setup Utility
DRAM Clock / Driv Control
Current FSB Frequency 100 MHz
Current DRAM Frequency 200 MHz
DRAM Clock [200 MHz]
Item Help
Menu Level ▶▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

The menu is to include settings for the chipset dependents features.These features are related to system performance. Make sure you fully understand the items contained in this menu before you change anything. You may change the parameter settings to improve system performance. However, it may make your system unstable if the setting is not correct for your system configuration.

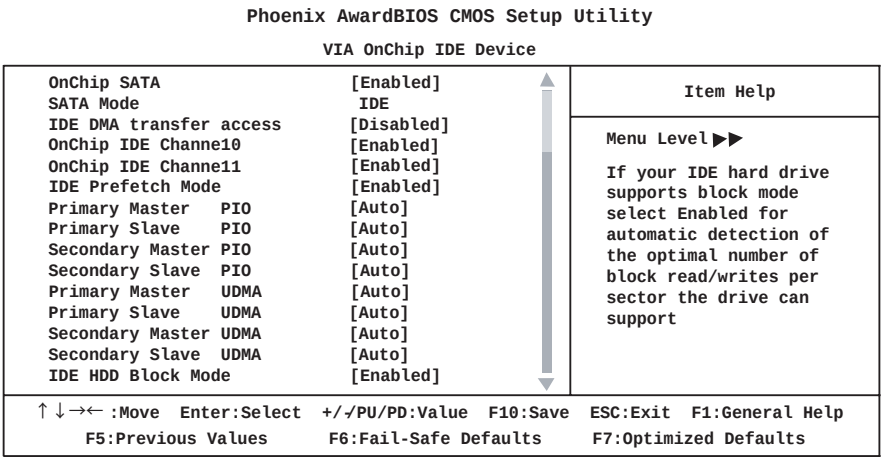
4-6-2 AGP & P2P Bridge Control


Phoenix AwardBIOS CMOS Setup Utility
AGP & P2P Bridge Control
VGA Share Memory Size [64M]
Direct Frame Buffer [Enabled]
Select Display Device CRT
Item Help
Menu Level ▶▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

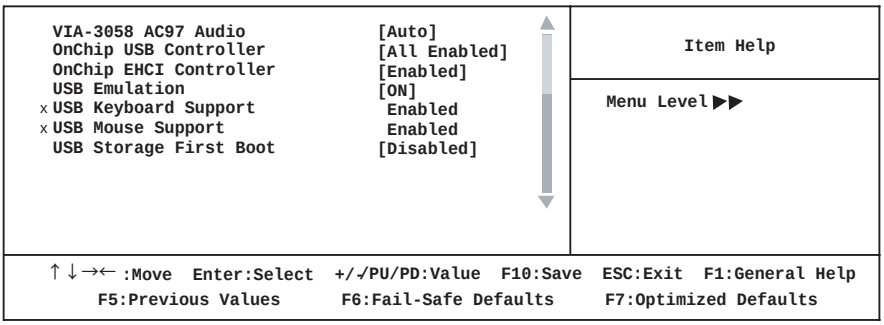
4-7 Integrated Peripherals


Phoenix AwardBIOS CMOS Setup Utility
Integrated Peripherals
▶VIA OnChip IDE Device [Press Enter]
▶VIA OnChip PCI Device [Press Enter]
▶Super IO Device [Press Enter]
Item Help
Menu Level ▶▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

4-7-1 VIA OnChip IDE Device


Phoenix AwardBIOS CMOS Setup Utility
VIA OnChip IDE Device
OnChip SATA [Enabled]
SATA Mode IDE
IDE DMA transfer access [Disabled]
OnChip IDE Channe10 [Enabled]
OnChip IDE Channe11 [Enabled]
IDE Prefetch Mode [Enabled]
Primary Master PIO [Auto]
Primary Slave PIO [Auto]
Secondary Master PIO [Auto]
Secondary Slave PIO [Auto]
Primary Master UDMA [Auto]
Primary Slave UDMA [Auto]
Secondary Master UDMA [Auto]
Secondary Slave UDMA [Auto]
IDE HDD Block Mode [Enabled]
Item Help
Menu Level ▶▶
If your IDE hard drive supports block mode select Enabled for automatic detection of the optimal number of block read/writes per sector the drive can support
↑↓→← :Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

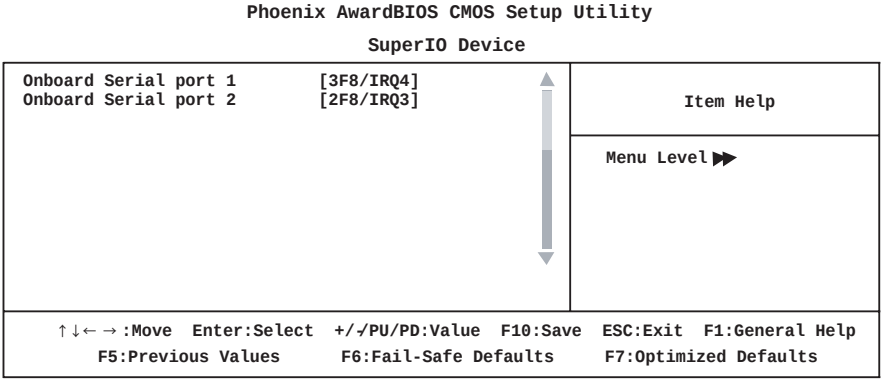
4-7-2 VIA OnChip PCI Device


Phoenix AwardBIOS CMOS Setup Utility
VIA OnChip PCI Device
VIA-3058 AC97 Audio [Auto]
OnChip USB Controller [All Enabled]
OnChip EHCI Controller [Enabled]
USB Emulation [ON]
x USB Keyboard Support Enabled
x USB Mouse Support Enabled
USB Storage First Boot [Disabled]
Item Help
Menu Level ▶▶
↑↓→← :Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

VIA-3058 AC97 AUDIO

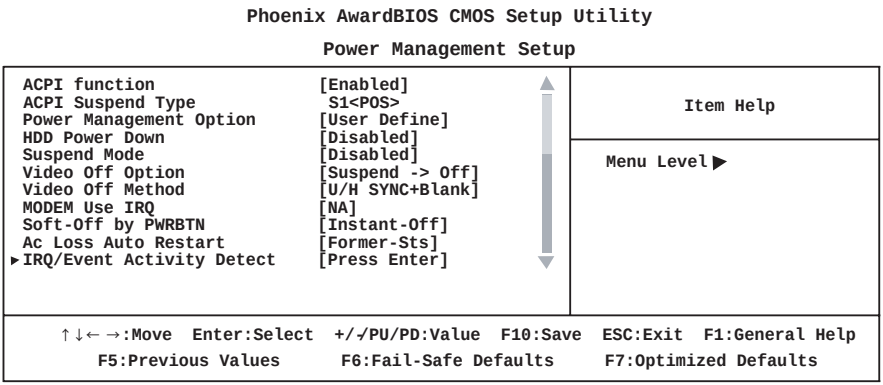
This item allows you to decide to enable/disable to support AC97 Audio.

4-7-3 Super IO Device


Phoenix AwardBIOS CMOS Setup Utility
SuperIO Device
Onboard Serial port 1 [3F8/IRQ4]
Onboard Serial port 2 [2F8/IRQ3]
Item Help
Menu Level ▶▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

4-8 Power Management Setup

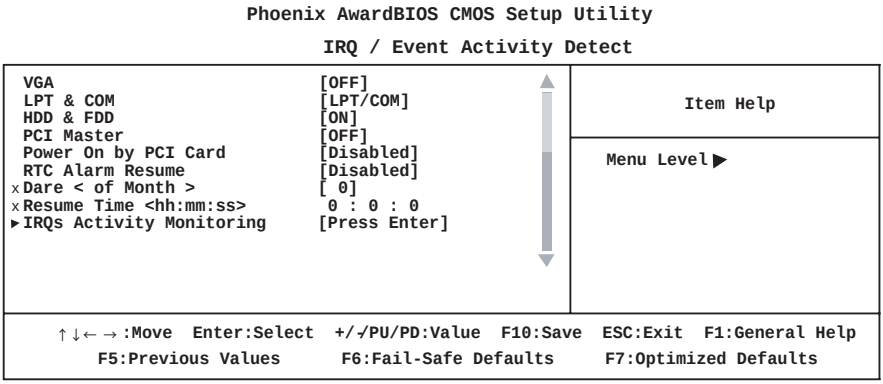
The Power Management Setup allows you to configure your system to most effectively energy saving while operating in a manner consistent with your own style of computer use.


Phoenix AwardBIOS CMOS Setup Utility
Power Management Setup
ACPI function [Enabled]
ACPI Suspend Type S1<POS>
Power Management Option [User Define]
HDD Power Down [Disabled]
Suspend Mode [Disabled]
Video Off Option [Suspend -> Off]
Video Off Method [U/H SYNC+Blank]
MODEM Use IRQ [NA]
Soft-Off by PWRBTN [Instant-Off]
Ac Loss Auto Restart [Former-Sts]
▶IRQ/Event Activity Detect [Press Enter]
Item Help
Menu Level ▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

IRQ / Event Activity Detect

Please refer to section4-8-1

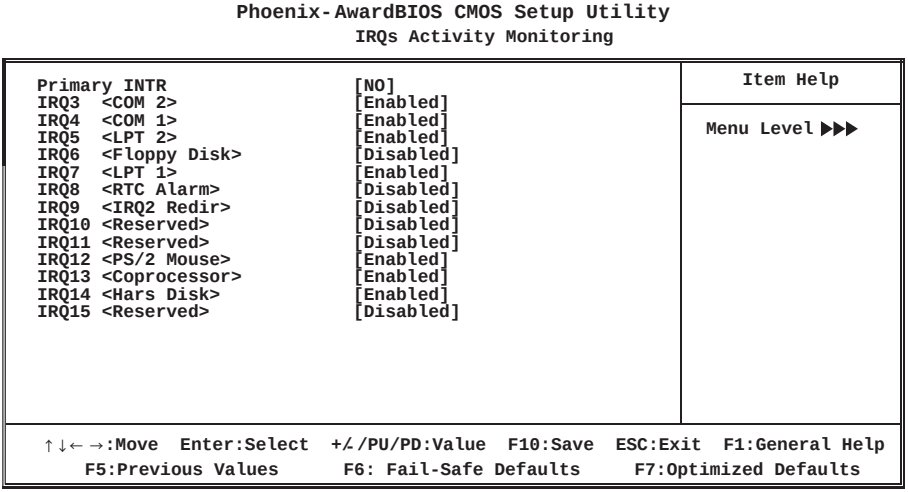
4-8-1 IRQ / Event Activity Detect


Phoenix AwardBIOS CMOS Setup Utility
IRQ / Event Activity Detect
VGA [OFF]
LPT & COM [LPT/COM]
HDD & FDD [ON]
PCI Master [OFF]
Power On by PCI Card [Disabled]
RTC Alarm Resume [Disabled]
x Dare < of Month > [0]
x Resume Time <hh:mm:ss> 0 : 0 : 0
▶ IRQs Activity Monitoring [Press Enter]
Item Help
Menu Level ▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6:Fail-Safe Defaults F7:Optimized Defaults

IRQs Activity Monitoring

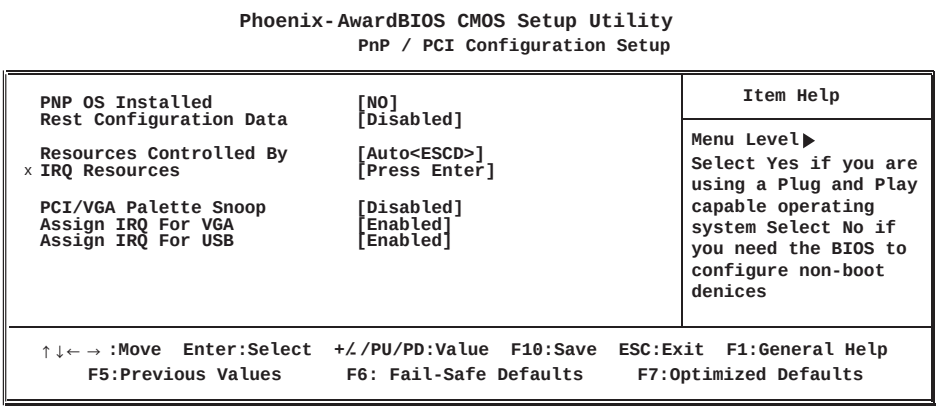
Please refer to section4-8-1.1

4-8-1.1 IRQs Activity Monitoring



4-9 PnP / PCI Configuration Setup

This section describes how to configure the PCI bus system. PCI, or Personal Computer Interconnect, is a system which allows I/O devices to operate at the speed the CPU itself keeps when CPU communicates with its own special components. This section covers some very technical items and we strongly recommended that only experienced users should make any change to the default settings.


Phoenix-AwardBIOS CMOS Setup Utility
PnP / PCI Configuration Setup
PNP OS Installed [NO]
Rest Configuration Data [Disabled]
Resources Controlled By [Auto<ESCD>]
x IRQ Resources [Press Enter]
PCI/VGA Palette Snoop [Disabled]
Assign IRQ For VGA [Enabled]
Assign IRQ For USB [Enabled]
Item Help
Menu Level▶
Select Yes if you are using a Plug and Play capable operating system Select No if you need the BIOS to configure non-boot denices
↑↓←→ :Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6: Fail-Safe Defaults F7:Optimized Defaults

Resource Controlled By

The Award Plug and Play BIOS can automatically configure all of the boot and Plug and Play compatible devices. However, this capability means absolutely nothing unless you are using a Plug and Play operating system such as Windows 95/98. If you set this field to "manual", choose a specific resource by going into each sub menu that follows this field (a sub menu is preceded by a ">").
The settings are Auto (ESCD) or Manual.

IRQ Resources

When resources are controlled manually, each system interrupt is assigned a type, depending on the type of device using the interrupt.

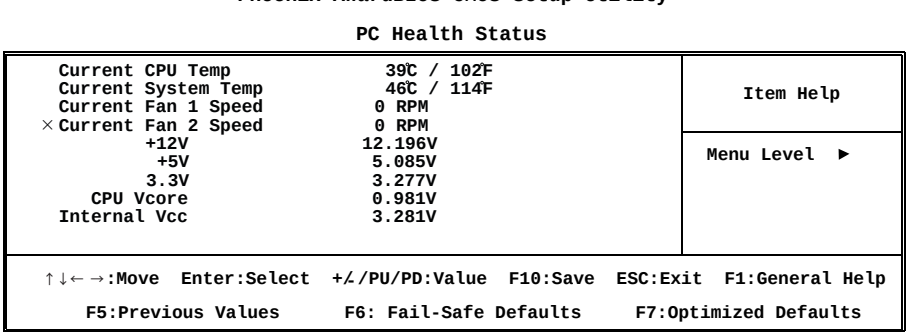
PCI/VGA Palette Snoop

Leave this field at Disabled. The settings are Enabled or Disabled.

4-10 PC Health Status

This section shows the status of your CPU, Fan, and overall system.

This is only available when there is Hardware Monitor function onboard.


Phoenix-AwardBIOS CMOS Setup Utility
PC Health Status
Current CPU Temp 39°C / 102°F
Current System Temp 46°C / 114°F
Current Fan 1 Speed 0 RPM
Current Fan 2 Speed 0 RPM
+12V 12.196V
+5V 5.085V
3.3V 3.277V
CPU Vcore 0.981V
Internal Vcc 3.281V
Item Help
Menu Level ▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6: Fail-Safe Defaults F7:Optimized Defaults

Show PC Health in Post

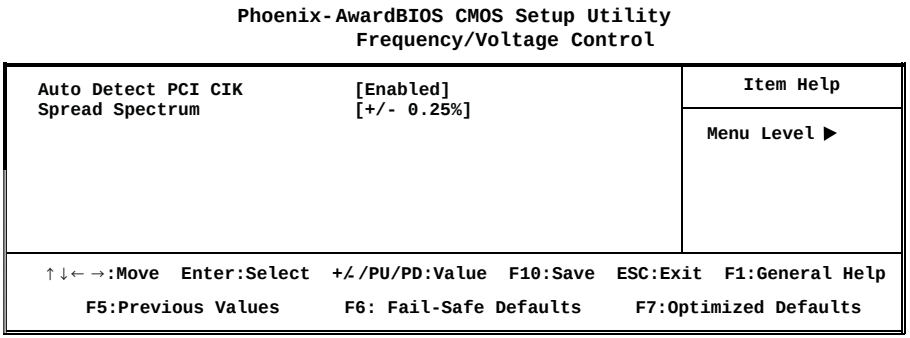
During Enabled, it displays information list below. The choice is either Enabled or Disabled

Current CPU Temperature/Current System Temp/Current FAN1,FAN2 Speed/ Vcore/3.3V/+5V/+12V

This will show the CPU/FAN/System voltage chart and FAN Speed

4-11 Frequency/Voltage Control

This section is to set CPU Frequency Control.


Phoenix-AwardBIOS CMOS Setup Utility
Frequency/Voltage Control
Auto Detect PCI CIK [Enabled]
Spread Spectrum [+/- 0.25%]
Item Help
Menu Level ▶
↑↓←→:Move Enter:Select +/-/PU/PD:Value F10:Save ESC:Exit F1:General Help
F5:Previous Values F6: Fail-Safe Defaults F7:Optimized Defaults

Spread Spectrum

This item allows you to set the CPU Host/PCI clock Spread Spectrum.

The choice are Disabled or Enabled.

4-12 Load Standard/Optimized Defaults

Load Standard Defaults

When you press <Enter> on this item, you get confirmation dialog box with a message similar to:

Load Standard Defaults (Y/N)? N

Press <Y> to load the BIOS default values for the most stable system operation and minimal performance.

Load Optimized Defaults

When you press <Enter> on this item, you get a confirmation dialog box with a message similar to:

Load Optimized Defaults (Y/N)? N

Press <Y> to load the default values that are factory settings for optimal system operation performance.

4-13 Set Supervisor/ User Password

You can set supervisor password, user password, or both. The differences are:

Supervisor password: You can enter the setup menus and change the options.

User password: You can enter the setup menus but do not have the right to change the options. When you select this function, the following message will appear at the center of the screen to assist you in creating a password.

ENTER PASSWORD:

Type the password, up to eight characters in length, and press <Enter>. The password typed will clear any previous password from CMOS memory. You will be asked to confirm the password. Type the password again and press <Enter>. You may also press <Esc> to abort the selection without entering password.

To disable a password, just press <Enter> when you are prompted to enter the password. A message will confirm if you want to disable the password. Once the password is disabled, the system will boot and you can enter Setup menus freely.

PASSWORD DISABLED.

When a password has been enabled, you have to enter it every time before you enter the Setup. This prevents an unauthorized person from changing any part of your system configuration.

Additionally, when a password is enabled, you can also have the BIOS to request a password at every time when your system is rebooted. This would prevent the unauthorized user.

You can determine if the password is required within the BIOS Features Setup Menu and its Security option. If the Security option is set to "System", the password will be required both at boot and at entry to Setup. If set to "Setup", the password is required only at the entry to Setup.

Chapter 5

DRIVER INSTALLATION

There is a SYSTEM INSTALL CD disk in the package. This CD has all the drivers you need and some free application programs and utility programs. In addition, this CD also includes an auto-detect software which can tell you which hardware is installed and which driver is needed so that your system can function properly.
We call this auto detect software SYSTEM INSTALL.

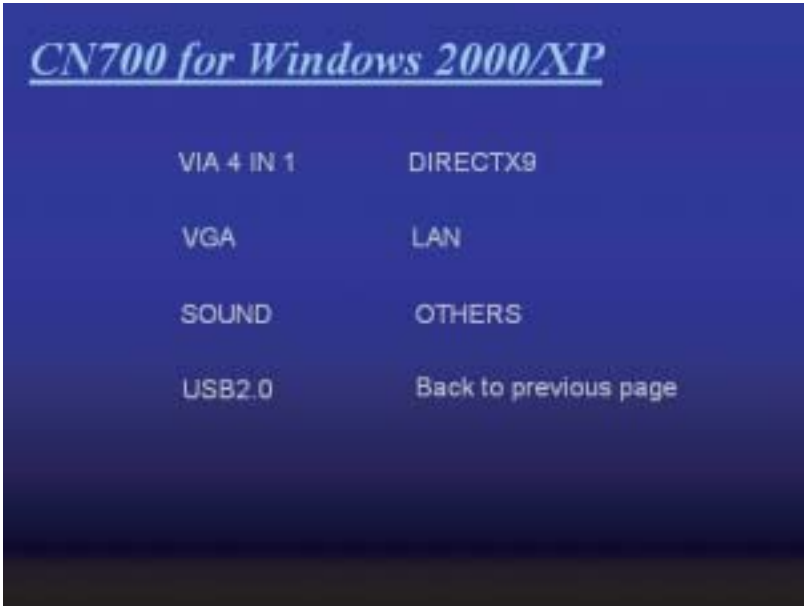
SYSTEM INSTALL Supports WINDOWS 98/98SE/ME/2000/XP

Insert the CD into your CD-ROM drive and the SYSTEM INSTALL Menu should appear as below. If the menu does not appear, double-click MY COMPUTER and double-click CD-ROM drive or click START, click RUN, and type X:\AUTORUN.EXE
(assuming X is your CD-ROM drive).

From SYSTEM INSTALL MENU you may make 3 selections:

1 . Auto detect main board and OS Into auto install driver Menu
2 . Explore CD to explore the contents of the CD
3 . EXIT to exit from SYSTEM INSTALL menu

Auto install driver Menu



1. VIA 4 IN 1 install VIA IDE/AGP/IRQ ROUTING/INF driver
2. VGA install on-board VGA driver
3. SOUND install AC97 Audio Codec Installing driver
4. USB2.0 install VIA USB2.0 hardware Installing driver
5. LAN to LAN install driver readme file
6. DIRECTX9 To install DirectX9 driver readme file
7.OTHERS To PenMount 6000 install driver readme file

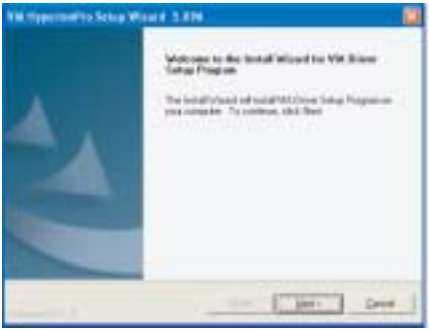
Note: PenMount 6000 DRIVER IS TO BE INSTALLED UNDER WIN 2000/XP ONLY.
Each selection is illustrated as below:

5-1 VIA 4 IN 1 Install VIA IDE/AGP/ INF Driver

INF :
Update Definitions of VIA chipsets for loading proper system driver.

PATA / SATA IDE :
Provide DMA capability for VIA PATA/SATA/PATA+SATA IED Controllers.

1.Click VIA 4 IN 1 when System Install MENU appears.

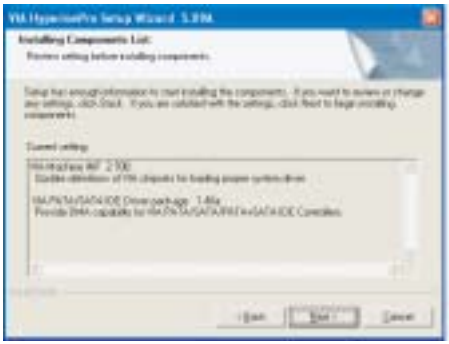


2 .Click NEXT when VIA Service Pack Wizard appears.

3. License Agreement, Click NEXT.

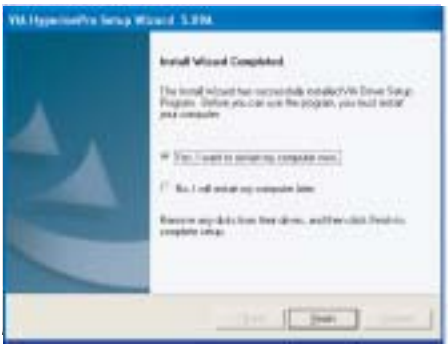
4. Choose all the drivers. Click NEXT.

Note: VIA AGP Driver IS TO BE INSTALLED UNDER WIN98 ONLY.



5. Installing Components List. Click NEXT.

6. This is to announce the Copy Right. Click NEXT.



7. Click Finish to restart computer.

NOTE: SYSTEM INSTALL will auto detect file path
X:\driver\VIA\CN700\4IN1\setup.exe
This driver supports WINDOWS 95\98\98SE\NT4.0\2000\XP

5-2 VGA INSTALL CN700 VGA Driver

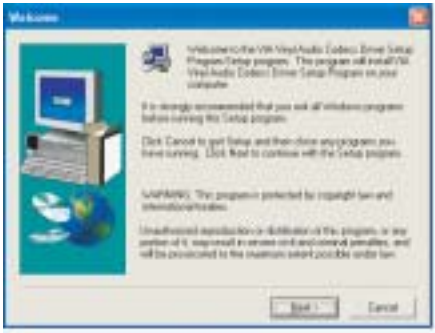
1.Click VGA when System Install MENU appears.

2.Click FINISH to Restart Computer.

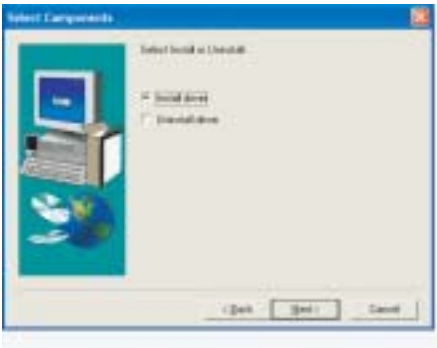
NOTE: The path of the file
For WINDOWS 98/98SE/ME
X:\driver\VIA\CN700\VGA\WIN98me\SETUP.EXE
For WINDOWS 2000/XP
X:\driver\VIA\CN700\VGA\WIN2KXP\SETUP.EXE

5-3 SOUND Install VIA Audio Codec Driver for VIA

1.Click SOUND when System Install MENU appears.



2.Click NEXT install VIA AC97 Audio Driver



3.Click NEXT install VIA AC97 Audio Driver

4.Click FINISH to Restart Computer.

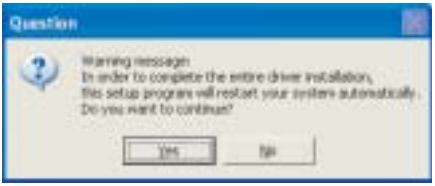
NOTE: The path of the file
For WIN98/ME/2K\XP
X:\ driver\via\viaudio\setup.exe

5-4 USB2.0 install VIA USB2.0 Installing driver

1.Click USB2.0 when System Install MENU appears.

2..Click NEXT install VIA USB2.0 Driver

NOTE:
When you install driver for USB2.0 under Windows 2000.Please install SP4 first.
When you install driver for USB2.0 under Windows XP.Please install SP1 first.



3.Click Yes to Install Driver and Restart Computer.

NOTE: The path of the file
For WIN98/ME/2K\XP
x:\ driver\via\CN700\USB20\setup.exe

5-5 HOW TO DISABLE ON-BOARD SOUND

Enter BIOS SETUP. Choose INTEGRATE PERIPHERALS. ChooseON-CHIP DEVICE FUNCTION. Choose AC97 AUDIO Disable on-board sound function by pressing PAGE DOWN KEY to Disable.

5-6 HOW TO UPDATE BIOS

In DOS Mode

STEP 1. Prepare a boot disc. (you may make one by clicking START, clicking RUN, typing SYS A: , and clicking OK)

STEP 2. Copy utility program to your boot disc. You may copy it from DRIVER CD X:\Diver\bios\AWDFLASH.EXE or download it from our web site.

STEP 3. Copy the latest BIOS for 3V700D from our web site to your boot disc.

STEP 4. Insert your boot disc into A:
Start the computer and type "Awdflash A:\3V700Dxx.BIN/SN/PY/CC/R"
3V700Dxx.BIN is the file name of the latest BIOS.
It may be 3V700DA1.BIN or 3V700DA2.BIN
SN means don't save the current BIOS data
PY means renew the current BIOS data
CC means clear the current CMOS data
R means restart computer

STEP 5. Press ENTER and the BIOS will be updated, computer will restart automatically.

Appendix A: Power Consumption Test

Condition

Item	Spec
CPU	VIA Eden 1Ghz
DDR RAM	DDR2 667 / 1G
Operating System	Windows XPP/SP3
Test Program	3D Mark 2001SE
HDD 3.5" STAT	Standard HDD
HDD 2.5" STAT	Slim Type HDD
Mini PCI Card	Wireless LAN IEEE802.11g

Test Result for reference !

Hard Disk	Stand by	Start up		Operation	Shut down
		Maximum	Stable	Maximum	Maximum
Standard HDD	0.05A	2.87A	1.15A	1.61A	1.35A
Slim Type HDD	0.05A	1.18A	0.82A	1.33A	1.08A

The power consumption depends on your device choice!

Appendix B: Resolution list

640 x 480 x (256 / 16bit / 32bit)
800 x 480 x (256 / 16bit / 32bit)
800 x 600 x (256 / 16bit / 32bit)
848 x 480 x (256 / 16bit / 32bit)
1024 x 576 x (256 / 16bit / 32bit)
1024 x 768 x (256 / 16bit / 32bit)
1280 x 720 x (256 / 16bit / 32bit)
1280 x 768 x (256 / 16bit / 32bit)
1280 x 800 x (256 / 16bit / 32bit)
1280 x 1024 x (256 / 16bit / 32bit)
1360 x 768 x (256 / 16bit / 32bit)
1366 x 768 x (256 / 16bit / 32bit)
1440 x 900 x (256 / 16bit / 32bit)
1600 x 1200 x (256 / 16bit / 32bit)
1680 x 1050 x (256 / 16bit / 32bit)
1920 x 1080 x (256 / 16bit / 32bit)
1920 x 1200 x (256 / 16bit / 32bit)
1920 x 1440 x (256 / 16bit / 32bit)